An Expert System for Load Flow Planning

by

Farid Yousef Abdul-Rahman Al-Hariri

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

June, 1994

Order Number 1360388

An expert system for load flow planning

Al-Harari, Farid Yousef Abdul-Rahman, M.S.

King Fahd University of Petroleum and Minerals (Saudi Arabia), 1994

U·M·I
300 N. Zeeb Rd.
Ann Arbor, MI 48106

AN EXPERT SYSTEM FOR LOAD FLOW PLANNING

BY

FARID YOUSEF ABDUL-RAHMAN AL-HARARI

A Thesis Presented to the

FACULTY OF THE COLLEGE OF GRADUATE STUDIES

KING FAHD UNIVERSITY OF PETROLEUM & MINERALS

DHAHRAN, SAUDI ARABIA

In Partial Fulfillment of the
Requirements for the Degree of

MASTER OF SCIENCE

In

ELECTRICAL ENGINEERING

JUNE, 1994

KING FAHD UNIVERITY OF PETROLEUM AND MINERALS

DHAHRAN 31261, SAUDI ARABIA

COLLEGE OF GRADUATE STUDIES

This thesis, written by Farid Yousef Al-Harari under the direction of his Thesis Committee, has been presented to and accepted by the Dean of the College of Graduate Studies, in partial fulfillment of the requirements for the degree of MASTER OF SCIENCE.

Thesis Committee

DR. ABDALLAH AL-SHEHRI
Thesis Advisor

DR. IBRAHIM EL-AMIN
Member

DR. MOSTAFA AREF
Member

DR. GEORGE OPOKU
Member

DR. ABDALLAH AL-SHEHRI
Department Chairman

DR. ALA AL-RABEH
Dean of Graduate Studies

2.7.94

Date

This thesis is dedicated to my family

ACKNOWLEDGEMENTS

The full support and assistance received from King Fahd University of Petroleum and Minerals and Saudi Consolidated Electric Company in the Eastern Province (SCECO-East) are greatly acknowledged.

I wish to express my appreciation with thanks to my advisor Dr. Abdallah M. Al-Shehri for his great support and cooperation. I also wish to thank Dr. Ibrahim El-Amin, Dr. Mostafa Aref and Dr. George Opoku members of my thesis committee for their efforts and cooperation.

I would like further to thank Mr. Anwar S. Al-Abdullah, Manager of Engineering & Design Services Department at SCECO-East for his full support and encouragement. The assistances and helpful information provided by Mr. Fahd Al-Saho, Manager of Systems Operation Department and Mr. Ray Fong, Chief Power Dispatcher at SCECO-East are greatly appreciated.

Finally I would like to thank my father, wife, brothers, and sisters for their patience, forbearance and support.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

خلاصة الرسالة

اسم الطالب : فريد يوسف عبد الرحمن الهرري

عنوان الدراسة : نظام الخبير لتخطيط تدفق الحمل

التخصص : هندسة كهربائية - نظم الطاقة

تاريخ الشهادة : يونيو ١٩٩٤م

موضوع الرسالة الرئيسي هوتطوير نظام الخبير لتخطيط تدفق الحمل في نظم الطاقة الكهربائية . أهم أهداف النظام المطور هو تحسين أداء وإبداع كلاً من الخبير والمستخدم المبتدئ في تخطيط العمليات في نظم الطاقة الكهربائية .

يستطيع النظام بمراجعة النتائج الإبتدائية لتدفق الأحمال في الشبكة الكهربائية وإكتشاف الأوضاع الغير طبيعية وتقديم إقتراحات لإجراءات ضبط ضرورية لتخفيف زيادة الأحمال في خطوط النقل وتعديل الجهد في مواقع الأحمال وكذلك تعديل في توليد الطاقة المعدمة وزاوية الجهد في مواقع المولدات إلى حدود تشغيلية .

فقد تم تطوير نظام الخبير المعد لتخطيط تدفق الحمل بإستخدام طريقة الأسس القانونية والأداة المسماه بـ " كلبس " والمبينة على التقنية المعروفة بـ " التسلسل الأمامي " . وتم إختبار قدرة النظام المطور على عدة نظم كهربائية وكانت النتائج مرضية .

درجة الماجستير في العلوم

جامعة الملك فهد للبترول والمعادن

الظهران - المملكة العربية السعودية

يونيو ١٩٩٤م

THESIS ABSTRACT

FULL NAME OF STUDENT	:	FARID YOUSEF AL-HARARI
TITLE OF STUDY	:	AN EXPERT SYSTEM FOR LOAD FLOW PLANNING
MAJOR FIELD	:	POWER SYSTEM
DATE OF DEGREE	:	June, 1994

A sensitivity based Intelligent Load Flow Planning Engine is developed. The main objective of the engine is to provide both an expert and novice user with a friendly working environment, aiming at enhancing a user's creativity in making operational plans in power system.

The developed engine can review the initial load flow results, detect abnormals and suggest necessary control measures to alleviate line overloads and adjust the load bus voltage, var generation and voltage angle of generator bus to desired operating limits.

The expert system is developed using Expert System tool called CLIPS based on rule-based method and forward chaining reasoning process. Simulation studies with the developed expert system applied to various power systems show satisfactory results.

MASTER OF SCIENCE DEGREE

KING FAHD UNIVERSITY OF PETROLEUM AND MINERALS

DHAHRAN, SAUDI ARABIA

JUNE, 1994

CHAPTER 1

INTRODUCTION

1.1 GENERAL

The behavior of modern power systems has become more and more complicated, causing the decision-making process to the power system operators to be increasingly difficult. In recent years the development of Expert Systems (ES) for power system operation, control and planning has been an active area of research. The ES developed to incorporate the existing knowledge in planning and operation can offer assistance to system operators to make a correct, timely decision during an unforeseen event.

Most of the applications of expert systems to power systems have so far been limited to "fault diagnosis", "restoration planning" and "alarm processing" [1, 2]. There are several expert systems applied to Voltage and Var Control, and most of them are prototypes for demonstration, research or field tests [1, 3]. So far there is only one expert system applied to electric power generation re-scheduling (shifting) to eliminate line overloads under system contingencies. However, ES application for voltage angle adjustment of generation buses to keep the angles within the transient stability limits is limited.

In this thesis the following Expert Systems, applied to power system are developed in one package.

1) ES for Corrective Re-Scheduling in Power Generation

The ES is capable of identifying those overloaded lines under normal operations or contingencies and to perform the required shift in real power generation to eliminate overloads on transmission lines.

2) **ES for Voltage and Var Adjustment**

The ES is capable of detecting the load buses whose voltage magnitudes violate the normal operating limits or the generator buses whose generated reactive powers exceed the generation limits. It provides corrective actions via existing control elements in the power systems, such as shunt reactors, shunt capacitors, transformer tap changers and generator voltages.

3) **ES for Voltage Angle Adjustment**

The ES can verify the voltage angle of a generator bus and can compare the difference between its value and the value of slack bus voltage angle, with the pre-defined limit for transient stability. It then performs shifts in real power generation to adjust the angle of the generator bus if the value found exceeds the limit.

The thesis is divided into five chapters. The Chapter 1 is an introductory part where the application of ES to electrical power systems, in general, and to load flow planning in particular is addressed. This is followed by definition of the problem of load flow planning in modern power systems. The chapter concludes by highlighting the scope of work for this thesis. The Chapter 2 discusses the expert system components, roles and advantages of the ES in power system applications, programming languages and tools. The last part of the chapter discusses in details the features of the selected ES programming shell, 'CLIPS'. The architecture and methodologies applied for building the load flow planning expert system are explained in Chapter 3. In Chapter 4, several case studies are provided to demonstrate the operation and capabilities of the developed ES prototype. Finally, Chapter

5 concludes the thesis with a summary and recommendations for future work.

1.2 LITERATURE REVIEW

Artificial Intelligence (AI) has been an active area of research in recent years. Significant progress has been made in the development of the expert systems. Early application of expert systems were in medical diagnosis and therapy, and computer system configuration and troubleshooting [3, 5-8]. Other applications include analysis and synthesis of electric circuits [5-8], computer aided control system design [6, 8, 9] and substation grounding design [9]. Application of expert systems to power systems is a new area. Most of the present expert systems in power systems are prototypes for demonstration, research or field tests [1, 4].

The present expert system applications in power system operation have included the following major aspects:

1) Fault Diagnosis of Network

Several expert systems have been developed to identify the initiating disturbance and any device misoperations involved in an accidental change in the configuration of a network [7, 10-12]. For example in reference [10] the authors developed an expert system, using PROLOG, which can estimate fault sections using information from protective relays and circuit breakers. In reference [11] the authors developed a distributed expert system for fault diagnosis based on what they called a hybrid approach and using a rule-based language (OPS5).

2) Alarm Processing

Intelligent Alarm Processors (IAP) have been proposed and developed for power control centers [7, 13-15]. The main objective of the IAP is to analyze the huge number of alarm messages produced by SCADA/EMS system and display the message on the dispatcher's console in different categories (i.e. most urgent alarms, less urgent alarms, non-urgent alarms).

3) Switching Operation

Expert systems for switching operations have been developed as prototypes [7,16]. The main function of the system is to give instructions to operators about how to switch on or off transmission lines during faults.

4) System Restoration

The expert system application to system restoration was pioneered by Sakaguchi and Matsumoto [7]. The knowledge base of their system contains rules which can identify paths of energization for de-energized buses. A more advanced expert system prototype, called a "guidance" system for restoration, was developed by Kojima, Warashia, et al [7]. Their system has been able to select the target system during the restoration, accept overriding actions and provide guidance in switching actions and load dispatching. The prototype developed by Liu, Lee and Venkata [17] performs restorations in distribution systems.

5) Substation Automation

B. Don Russell and Karan Watson [18] described in their paper the various substation automation functions where knowledge base systems could ideally be used. The paper includes justifications and preliminary field experiments.

6) Load Flow Planning

* In reference [19] the authors developed an intelligent load flow engine for power system planning. Their expert system identifies the overloaded lines and buses with violating voltage levels and provide recommendations for adjusting the active load flow and bus voltage.

Once the expert system identifies the overloaded lines and under/over voltage buses, it starts searching heuristically to solve the problem in two steps. In the first step, adjustment of active power flow is carried out. This is done by adjusting the generator outputs. While in the second step the bus voltages are adjusted using VAR compensators distributed in a network such as a shunt reactor, static condenser, tap changer, etc. The authors believe that the effect of VAR compensation is limited within a local area. Accordingly the electrical distance measured with the impedance between two buses was used as a clue to heuristic search. VAR compensators are examined in the order of distance. The amount of compensation is estimated with the sensitivity of $\Delta V/\Delta Q$ obtained from the DC model.

After the the adjustment is made, the load flow is computed again to see if the goal conditions are satisfied. DC load flow method was used to estimate line loadings and bus voltages.

* In reference [20, 21] the authors presented a preliminary procedure for developing sensitivity based intelligent load flow engine. Their attempt is to develop an engine which is capable of suggesting corrective measures for line overloads, load bus voltage and reactive power source limit violations.

7) Security Assessment

Dejan J. Sobajic and Y. Pao [22] developed a rule based system to effectively (1) detect and screen out harmless contingencies, and (2) recognize potentially harmful contingencies and determine corresponding endangered areas. The rule-based screening procedure is based on available topology-dependent qualitative information in power system and some equations to calculate the security margins. The expert system was tested on several power system models.

8) Reactive Power and Voltage Control

Expert systems have been developed to assist in the decision-making of the reactive power/voltage control problem. Hereunder, the different approaches of some expert systems in the reactive power/voltage control will be highlighted.

* The expert system, presented by Chen-Ching Liu and Kevin Tomsovic in references [7] and [8], consists of three fundamental components: a global database, a knowledge base, and an inference engine utilized to chain a set of rules to form a line of reasoning. A forward-chaining rule-based program (OPS83) was used for the execution of the expert system.

  The knowledge base of this expert system is composed of the following empirical rules:

  (a) If a load bus voltage drops below (or rises above) the operating limit, control devices such as shunt capacitors, transformer tap changers, generator voltage and synchronous condensers can be switched or adjusted to restore profile.

(b) If a load bus voltage drops below (or rises above) the operating limit, it is most efficient to apply the reactive compensation locally. If the capacity of local compensators is insufficient to resolve the voltage problem, then compensators with next highest sensitivities should be chosen.

(c) If the voltage is low (high) at a bus, the tap position of the local transformer tap changer can be raised (lowered) to correct the problem. However, increasing (decreasing) the tap position may cause other loads bus voltages to drop (rise).

(d) Generator bus voltages can be raised (lowered) to solve the low (high) load voltage problems.

The imperical rules are applied based on different levels of violations. In case of slight violation of the voltage limit, the above rules (a) - (d) are applied. However, if severe violation exists, a more systematic method such as a reactive power dispatch algorithm using linear programming is required to efficiently find a solution. The expert system was tested on a modified IEEE-30 bus system and the results presented are very encouraging.

* S.J. Chang, O.P. Malik and G.S. Hope [6] developed an expert system which also controls the voltage and reactive power of a power system.

The fundamental ideas for the development of the basic rules and the inference engine of the proposed expert system are outlined here.

In order to easily analyze the relationship between the bus voltage and the control measures, N_1 sensitivity trees, (See Figure 1), equal to the number of load buses in the power system, are used to represent system relationships. Each tree represents the entire relationship between a particular bus voltage and all control measures and secondary variations of the other bus voltage.

The root of the tree is the controlled bus voltage. Each root connects the control measures through branches. The sensitivity factor associated with the controlled bus voltage and the control measure is given on the corresponding branch. The second level of the tree gives the effect of the control measures on all other load bus voltages. In the second level, each control measure is connected to the other bus voltages by sub-branches. The sensitivity factors between the control and the bus voltages are shown at the sub-branches.

Similar trees are made for the reactive power dispatch control. In these trees, the reactive powers drawn at each load bus become the control measures. The roots and the second level leaves are the same as the voltage control sensitivity trees. Appropriate sensitivity factors corresponding to the reactive power dispatch and the bus voltage are used.

Based on the sensitivity tree, the basic rules and the inference engine of the proposed voltage and reactive power control expert system are developed. By recursively using predicate calculus rules in cooperation with

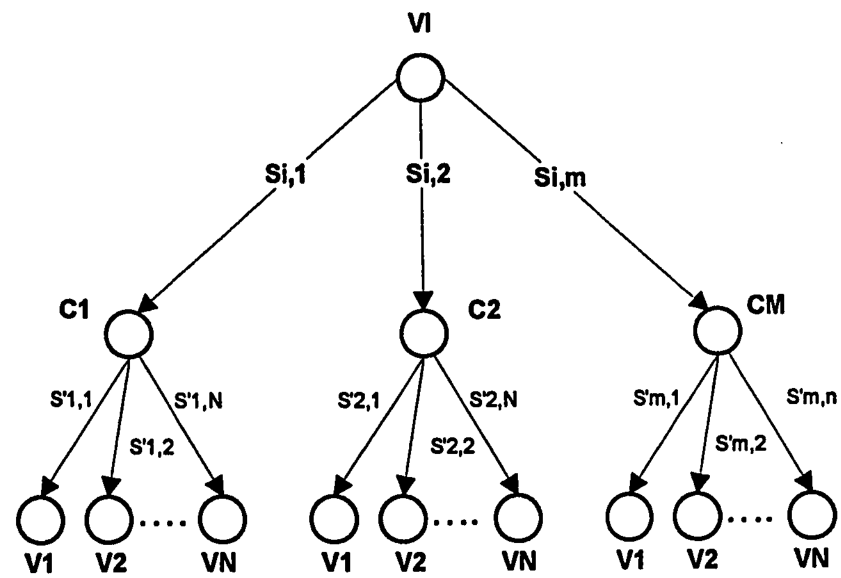

Figure 1: A Sensitivity Tree

123-6

numerical and logical calculations, the inference engine of the expert system was developed in PROLOG language which is based on backward chaining.

* Terry F. Godart and Hans B. Puttgen [23] developed a voltage control expert system based on two rule-based techniques which predict the impact of voltage control devices on power system bus voltages using a "reactive path concept". The first technique determines a set of buses on which a particular controller has a significant impact. The second technique determines the two reactive power compensators which are best suited to alleviate voltage violations at a particular bus. These two techniques are implemented as tools for the voltage control expert system using OPS/83.

The knowledge base of the first technique was developed based on an assumption that each compensator's impact is limited to a set of buses meaning to its Reactive Power Control (RPC) region. As a result, the operation of a controller is determined based on its control region alone.

RPC regions are determined using a heuristically guided breadth-first search among the power system buses and using the base-case system topology. Heuristics are based on base-case reactive power flow data.

As indicated above the purpose of the second technique is to identify the two reactive power compensators that are best suited to mitigate voltage violations at a particular bus designated as "observed" bus. This technique attributes to every observed bus a Most Efficient Controller (MEC) and a Second Most Efficient Controller (SMEC).

The MEC and SMEC for a given observed bus are determined using a heuristically guided depth-first search. A reactive path is built from the observed bus; the first compensator encountered which is directly connected to this path is designated as MEC whereas the second one is the SMEC.

The two rules based techniques were tested by the authors using a version of the IEEE-118 bus system and the results show that these techniques provide valid results and that they are more efficient and feasible within the framework of an expert system.

SUMMARY

Expert System technology has obtained wide ranging attention of the power engineering community, researchers at universities and research institutes, and operation staff of electric utilities. Both researchers and users have realized that many operation and planning problems in power system are ideally suitable for expert system applications.

Some interesting observations drawn from the literature survey are summarized as follows:

- An Expert System can implement the performances of a person with recognized expertise in the problem domain. Therefore, it can reduce tedious and redundant manual tasks and provide a human expert with an environment that improves his productivity thus leading to efficient operation,

- There is a class of problems for which the application of conventional software is not suitable and that could be instead rationally and efficiently solved by using the ES technology,

- o At end of 1993 the practical use of ES in power system operation was fairly modest,
- o Most of the applications of ES to power systems have so far been limited to "fault location", "restoration planning" and "alarm processing",
- o Expert Systems developed for "Remedial Control" and "Security Assessment" in power system are so far very few.

1.3 PROBLEM DEFINITION

The growth in size and complexity of power system networks has necessitated more sophisticated techniques to power system planning, operation and control.

The power system planning, operation and control tasks have so far been achieved partly by great efforts of experienced operators or planners and partly by the introduction of partially automated systems which have become possible by the active application of advanced analysis techniques. Despite the remarkable progress in the extent and quality of the automation technology, much still depends on the judgement of human experts.

At present, judgements under abnormal conditions in power systems are also left to human experts. Moreover, the anticipated shortage in the supply of sufficient experienced human resources in the near future makes it is imperative to further promote automation in power systems.

The Artificial Intelligence (AI), especially expert system applications, can be the alleviation of the bottlenecks which most electric utilities are currently facing in system planning, operation and control.

1.4 SCOPE OF WORK

The main objective of this thesis is to develop an interactive ES package, consisting of three independent Knowledge Bases, that will assist the power system operators and planners in the decision-making when performing load flow analysis. Specifically, it will perform the following tasks:

1) Corrective Re-Scheduling in Power Generation

The developed expert system should be able to review the load flow results and detect the overloaded lines in a power network and re-schedule the power generation to eliminate the overloads.

2) Bus Voltage and Reactive Power

The developed expert system should assist the user in detecting load bus voltage problems in power network and determine appropriate control actions to correct the detected problems. Also, it should control the reactive power flow in the power network.

3) Voltage Angle Adjustment

The ES should detect voltage angle violation for generation bus and determine appropriate new setting in power generation to eliminate the violation.

The ES package developed should be interfaced with other power system application programs, such as, load flow programs and power system sensitivity coefficient computation programs, written in FORTRAN.

The following items describe the scope of work followed to achieve the above objectives:

1) <u>Knowledge Acquisition</u>

The knowledge used in the applications was acquired from:

- o literature survey in the relevant domain,
- o discussion with experts in the power system dispatch center at SCECO-East, the electric utility responsible for generation, transmission and distribution of electricity in the Eastern Province of Saudi Arabia.

2) <u>Selection of ES Tool</u>

Different available AI languages and ES tools were evaluated and the selection was based on the following characteristics:

- o easy to learn
- o flexible for addition and deletion of rules
- o PC based
- o friendly and easy to use
- o simple in knowledge representation
- o capable to support forward-chaining control strategy.
- o capable to communicate and interface with other languages, such as FORTRAN and external files.

3) <u>Development of Power System Application Programs</u>

The following three programs were developed using FORTRAN language to determine the initial state and the sensitivity coefficients of the power system that are needed by the Knowledge Bases of the developed ES package:

- Load Flow Program
- Voltage, Angle and Var Sensitivity Coefficients Calculation Program.
- Line Power Flow Sensitivity Coefficients Calculation Program.

4) **Development of the ES Package Configuration**

A configuration which describes the interfaces among the different components of the ES, application programs, and input Data Bases and user was developed.

5) **Development of the Knowledge Base (KB)**

The KBs were developed from the knowledge gained from the literature survey and discussions with power dispatchers at SCECO-East. During the development of the KBs, the following main aspects were considered:

- varieties of control devices (continuous and discrete)
- priority order in control devices
- locality in control effects
- generality (not specific to a particular system)

6) **Testing:**

The developed ES package was tested on several power systems to demonstrate its capabilities.

CHAPTER 2

EXPERT SYSTEMS IN POWER SYSTEM OPERATION AND CONTROL

2.1 INTRODUCTION

In recent years, expert system technology has captured the interest of many electric engineers and this trend is likely to continue. One fact of the current trend is to apply Expert Systems (ES) to power systems. Expert systems have been developed for several power system applications such as system planning, operation and control.

This chapter is a broad introduction to expert systems. The definition and fundamental principles of expert systems are introduced. The role and advantages of expert systems in power system applications are discussed. The various expert systems programing languages and tools are discussed. Finally, the chapter presents the main features of the ES programing tool "CLIPS" which is used for the development of the intelligent load flow engine.

2.2 DEFINITION AND FUNDAMENTAL COMPONENTS OF THE ES

An Expert System is a branch of AI which behaves like a human expert in a narrow specified domain of application. Each system has a knowledge in a particular domain and it utilizes this to provide decision support at a level comparable to the human expert and is capable of justifying its reasoning [5-7, 24].

Any Expert System mainly consists of three fundamental components, i.e.

- a knowledge base
- an inference engine
- a user interface

The structure of an expert system in a general block diagram form is shown in Figure 2.

The knowledge base contains knowledge which is specific to the domain of application, such as, simple facts about the domain, rules that describe relations in the domain, possible methods, heuristics and idea for solving problems in the domain. The inference engine actively uses the knowledge in the base and performs as cognitive processor. It makes all the reasoning and tries to find the rules that will be activated by the appropriate stimuli. The user interface provides smooth communication between the user and the system. The inference engine and the interface generally considered as one module and it is called an expert system shell.

2.3 ROLE AND ADVANTAGES OF THE ES IN POWER SYSTEM APPLICATIONS

The behavior of modern interconnected power systems has become more and more complicated, causing sudden rise of security problems to electric utilities. Often, utilities face the problem that the number of operators with relevant experience is not sufficient to cope with structurally changing reliability. These facts may ask for user friendly, alert assistance by ES when the operator needs to understand the electric situation and to make a correct, timely decision during an unforeseen event. Similarly, some well known operating

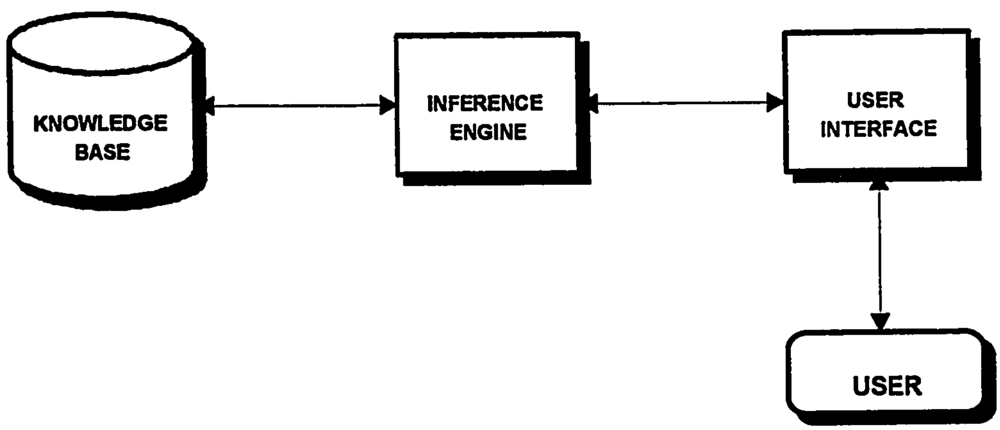

Figure 2: Block Diagram of Expert System

problems still have to be solved, such as: alarm handling, voltage/reactive power control, security assessment etc.

The role of the ES in power system operation could be [4]:

1) to work beside a human expert in high-level intellectual activities, such as: restoration, security analysis, operation planning, performance analysis;
2) to replace the human expert in medium - and low-level intellectual activities such as: diagnosis, data interpretation, maintenance coordination;
3) to support the man-machine interface;
4) to support the training activities;
5) to collect and organize knowledge in a specific domain (i.e planning) and then provide its maintenance and preservation;
6) to support decision on regulation and control functions specifically in critical complex situations;
7) to support human performance in monitoring;
8) to support managerial corrective action.

The main advantages of Expert Systems over numerical methods are [3, 8, 24]:

1) Assist Human Experts

During a real-time environment, if a problem is detected, an expert system can be applied to suggest a proper solution to the operators in time, based on the incorporated expert knowledge.

2) Understanding

The knowledge required to perform a task is expressed in terms of production rule (IF-THEN Structures), which are very close to natural language and, therefore, easy to understand by users.

3) **Flexibility**

Each production rule represents a piece of knowledge relevant to the task. Hence it is very convenient to add or remove a rule when more experience is gained during operation and testing.

4) **Universality**

The knowledge base is problem domain dependent, but the inference engine is domain independent. So, different expert systems can be developed by replacing the knowledge base.

5) **Rapidity**

The expert system can provide the right expertise whenever needed. Expert systems can provide more rapid decisions to emergency events than human operators. This is very useful in power system operation.

2.4 EXPERT SYSTEMS PROGRAMMING LANGUAGES AND TOOLS

Altough expert systems is a branch of AI, there are specialized languages for expert systems that are quite different from the commonly used AI languages such as LISP and PROLOG.

An expert system language is a higher order language than languages like LISP or C. That is, the specialized nature of expert system languages makes them very suitable for writing expert systems but not for general purpose programming.

The primary functional difference between expert system languages and procedural languages, such as, FORTRAN or ADA is the focus of representation. Procedural languages focus on providing flexible and robust techniques to represent data. For example, data structures such as arrays, records, linked lists and trees are easily created and manipulated. Also, the data and methods to manipulate the data are tightly interwoven in the procedural languages. In contrast, expert system languages focus on providing flexible and robust ways to represent knowledge.

This difference in focus also leads to a difference in program design methodology. Because of the light interweaving of the data and knowledge in procedural languages, programmers must carefully describe the sequence of execution. However, the explicit seperation of data from knowledge in expert system languages requires considerably less rigid control of execution sequence.

There are several expert system languages and tools commercially available. Examples of such tools supported for micro-computers are GURU, EXSYS, KES, LEVEL 5, CLIPS and few others. The main characteristics of these tools are summarized in Table 1 [24, 25].

TABLE 1: CHARACTERISTICS OF EXPERT SYSTEMS TOOLS FOR MICROCOMPUTERS

Name	EXSYS PROF-FESSIONAL	GURU	LEVEL 5	KES II	CLIPS
FUNCTIONAL USE	Wide range including diagnosis and real time processing	Variety including medical and personal application	Variety of applications	Variety	Variety
K.B. Represen-tation					
Rules	x	x	x	x	x
Object-oriented	x			x	x
Procedures	x (by calling out)	x	x	x	x
Inference Engine					
Forward Chaining	x	x	limited	x	x
Backward Chaining	x	x	x	x	x
Math Calculaions	x	x	x	x	
Pattern Matching				x	x
Developer Inter-face					
Word Processor	x	x	x	Exernal package	x
K.B. Editor	x	x		External	x
Menu	Commands	x	x		x
Inference Tracing	x	x	x	x	x
On-line Help	x	x	x	x	x
Syntax Help		x	x	x	x
Software					
Language of Tool	C	C	Pascal	C	C
Operating System	MSDOS, UNIX, VMS	MSDOS, VMS	MSDOS, VMS	MSDOS, XEXIX UNIX ULTRIX VMS, MVS	MSDOS WINDOWS MACINTOSH UNIX,VMS

Table 1 Continued.

Computer Support					
VAX	x	x	x	x	x
IBM PC	x	x	x	x	x
IBM PS/2	x	x		x	x
Macintosh	x				x
Sun	x			x	x
Appolo				x	x
IBM Mainframe				x	x
Others					
System Interface					
Language Hooks	Any Procedural Language	C	Any Procedural Language	Any Procedural Language	Any Procedural Language
D.B. Hooks	dBase	Own Rela-tnl/DB	dBase	dBase, IMS	
End User Interface					
Line		x	x	x	x
Menu	x	x	x	x	x
Windows					x
Company	EXSYS	mdbs	Levels(1B1)	Software A & E	NASA

2.5 'C' LANGUAGE INTEGRATED PRODUCTION SYSTEM (CLIPS)

CLIPS, the C Language Integrated Production System, is a complete environment for developing expert system programs which are specifically intended to model human expertise or knowledge. It is designed to allow artificial intelligence research, development, and delivery on conventional computers. CLIPS provides a cohesive tool for handling a wide variety of knowledge with support for three different programming paradigms: rule-based, object-oriented, and procedural. Rule-based programming allows knowledge to be represented as heuristics, or "rule-of-thumb" which specify a set of actions to be performed for a given situation. Object-oriented programming allows complex systems to be modeled as modular components (which can be easily reused to model other systems or create new components). The procedural programming capabilities provided by CLIPS allow it to represent knowledge in ways similar to those allowed in languages such as C, Pascal, Ada, and LISP. Using CLIPS, one can develop expert system software using only rule-based programming, only object-oriented programming, only procedural programming, or combinations of the three.

CLIPS provides extensive features to support the rule-based programming paradigm including seven conflict resolution strategies, dynamic rule priorities, and truth maintenance. CLIPS supports more complex nesting of conditional elements in the if portion of a rule ("and", "or", and "not" conditional elements can be placed within a "not" conditional element).

The CLIPS Object-Oriented Language (COOL) provides object-oriented programming capabilities. Features supported by COOL include classes with multiple inheritance, abstraction, encapsulation, polymorphism, dynamic binding, and message passing with message-handlers. CLIPS supports tight integration of the rule-based programming features of CLIPS with COOL (that is, a rule can pattern match on objects created using COOL).

CLIPS provides the capability to define functions, overloaded functions, and global variables interactively. In addition, CLIPS can be embedded within procedural code, called as a subroutine, and integrated with languages such as C, FORTRAN and Ada. CLIPS can be easily extended by a user through the use of several well-defined protocols. It provides several delivery options for programs including the ability to generate stand alone executable or to load programs from text or binary files.

CHAPTER 3

THE INTELLIGENT LOAD FLOW PLANNING ENGINE

3.1 INTRODUCTION

In this Chapter, the methodology and approach followed for developing the subject ES prototype are discussed in details. First, the overall configuration is discussed. Next, the theoretical concepts adopted for developing the power system application programs needed for successful operation of the developed ES prototype are explained. Finally, the knowledge bases and solution processes are discussed.

3.2 CONFIGURATION OF THE DEVELOPED ES PACKAGE

The configuration of the proposed expert system is shown in Figure 3. It consists of SIX fundamental components:

1) User interface: provides a friendly interactive windows interface environement which allows the user to select the commands through the built-in menu. The main menu contains options to run the built-in editor, used for editing the input data and viewing/printing the output results, the application program (Load Flow) and the various expert system modules, namely, Generation Rescheduling, Voltage/Var Control and Voltage Angle Adjustment.

2) Global Input Data Base: contains user input data required for expert system

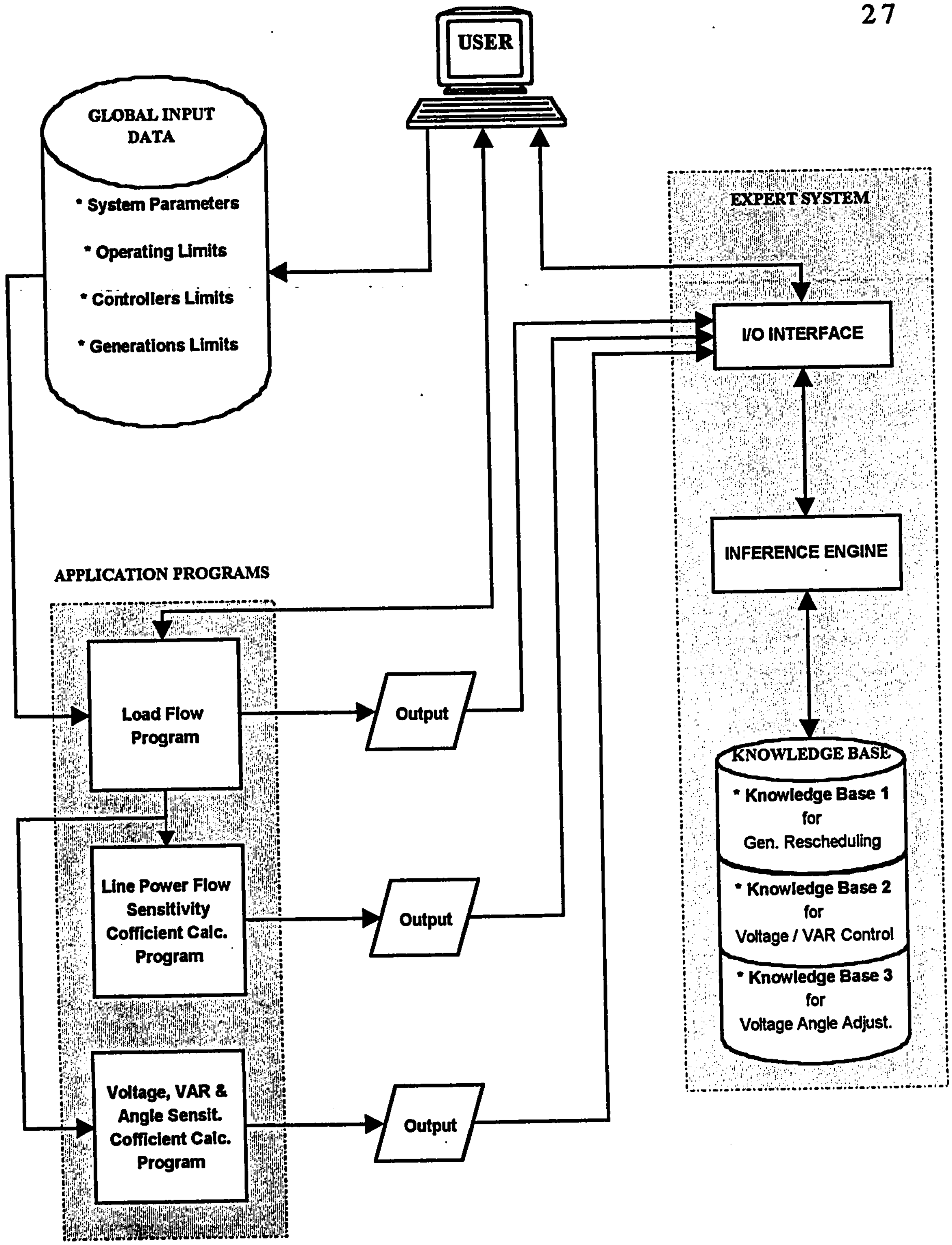

Figure 3: The Prototype Expert System Configuration.

and application programs. All the input data required for both the load flow program and the expert system is entered once and saved in a common file.

3) Input-Output interface: reads all the required input data (i.e. initial state of the power system, control elements and their limits etc.) saved in several files and save the output results of the various expert system modules.

4) Inference Engine: utilized to chain the facts given in the knowledge base and the basic rules in a forward chaining philosophy.

5) Knowledge Base: presents the facts and rules required to solve a problem.

- Knowledge Base 1: contains facts/rules for detection of overload lines and re-scheduling the generations to eliminate the overloads.

- Knowledge Base 2: contains facts/rules for detection of load bus voltage and generated VAR violations and selection of possible control measures to correct the detected violations.

- Knowledge Base 3: comprises of facts/rules to detect voltage angle violations and present remedial action by re-scheduling the generation.

6) Application Programs: used by expert system to perform necessary power system calculations, such as, load flow, sensitivity coefficients. They also, save all the input files needed for the expert system modules in required formats.

3.3 POWER SYSTEM APPLICATION PROGRAMS

Some power system application programs are developed using Fortran Language to assist the expert system performing its tasks. These programs are mainly:

- Load flow analysis program,
- Line power flow sensitivity coefficient calculation program,
- Voltage/Magnitude and Angle, and Var sensitivity coefficient calculation program.

3.3.1 Load Flow Program

Load flow calculations provide power flows and voltages of a network for specified terminal or bus conditions. This information is essential for the continuous evaluation of the current performance of a power system and for analyzing the effectiveness of alternative plans for system expansion or alternative control actions for maintaining the system under operating constrains. These analyses require the calculation of numerous load flows for both normal and emergency operating conditions. Therefore, a computer program is developed for load flow calculations.

The load flow program is developed using the Newton-Raphson method which uses bus admittances for solving the load-flow problem. Details on this method are given in reference [26]. The listing of the program is given in Appendix - A.

3.3.2 Line Flow Sensitivity Coefficients Calculation Program

Excessive load at a bus or increased generation in part of a network or outage of a line or a transformer in a power system may result in a number of element overloads. A shift of generation may alleviate this problem. Before shifting the generation, the relationship between the change in line flow and change in generation should be known.

The Line Flow Sensitivity Coefficients Calculation Program is developed to determine the sensitivity relationships between the power flows through elements and the real power injections at various generator buses. The relationships are determined using the triangular factors of the Jacobian matrix involved in the Newton-Raphson load flow studies and they are used by the developed ES package for finding the proper shift in generation to eliminate overloading of lines and transformers.

In this section the basic theory used for calculation of the sensitivity coefficients for the line flow is first discussed. Then the solution algorithm of the developed program is presented.

The listing of the Computer program is provided in Appendix - A.

Basic Theory

The basic theory presented here is given in reference [27]. Let a line connecting buses i and j be overloaded. Desired changes in line flows can be expressed in terms of change in power injections at generator buses. By definition, the overload value on the line from i to j is:

$$\Delta P_{ij} = P_{ij}\ (\text{rated}) - P_{ij}\ (\text{actual})$$

The overload is now related to changes in generation, at buses 2, 3 - - m:

$$\left[\Delta P_{ij}\right] = [H]\begin{bmatrix}\Delta P_2\\ \Delta P_3\\ \cdot\\ \cdot\\ \cdot\\ \Delta P_m\end{bmatrix} \tag{3-1}$$

where ΔP_2 , ΔP_3 , ... ΔP_m are the required generation shifts.

The matrix [H] can be expressed as:

$$[H] = \left(\frac{\partial P_{ij}}{\partial P}\right) = \left(\frac{\partial P_{ij}}{\partial P_2}\ \frac{\partial P_{ij}}{\partial P_3} \cdots \frac{\partial P_{ij}}{\partial P_m}\right)$$

Since $P_{ij} = f(\theta,V)$ and $(\theta,V) = f(P)$

$$[H] = \begin{bmatrix}\frac{\partial P_{ij}}{\partial \theta} & \frac{\partial P_{ij}}{\partial V}\end{bmatrix}\begin{bmatrix}\frac{\partial \theta}{\partial P}\\ \frac{\partial V}{\partial P}\end{bmatrix} \tag{3-2}$$

Determination of the [H] Matrix and its generalized inverse

The vectors $\frac{\partial P_{ij}}{\partial \theta}$ and $\frac{\partial P_{ij}}{\partial V}$ of equation (3-2), together will have a maximum of four non-zero elements for every P_{ij} , namely:

$$\frac{\partial P_{ij}}{\partial \theta_i},\ \frac{\partial P_{ij}}{\partial \theta_j},\ \frac{\partial P_{ij}}{\partial V_i},\ \text{and}\ \frac{\partial P_{ij}}{\partial V_j}$$

These partial derivatives can be calculated using the following expressions;

$$\frac{\partial P_{ij}}{\partial \theta_i} = - V_i V_j Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij})$$

$$\frac{\partial P_{ij}}{\partial \theta_j} = V_i V_j Y_{ij} \sin(\delta_i - \delta_j - \theta_{ij}) \tag{3-3}$$

$$\frac{\partial P_{ij}}{\partial V_i} = V_j Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$$

$$- 2V_j Y_{ij} \cos \theta_{ij}$$

$$\frac{\partial P_{ij}}{\partial V_i} = - V_i Y_{ij} \cos(\delta_i - \delta_j - \theta_{ij})$$

where

$V_i \angle \delta_i$; $V_j \angle \delta_j$ are the voltages of buses i and j, and

$Y_{ij} \angle \theta_{ij}$ is the negative of the line admittance between buses i and j.

The vectors $\frac{\partial \theta}{\partial P}$ and $\frac{\partial V}{\partial P}$ (equation (3-2)) are elements of the sensitivity matrix (inverse of Jacobian matrix). The triangular factors of the Jacobian matrix is used to determine the value of these elements.

Digital-Computer Solution Algorithm

The calculation of the line flow sensitivity coefficients is basically a two step process. First, the solution of the load flow for the power system is determined. Then, using the Jacobian matrix determined at the last iteration of the load flow program, the sensitivity coefficients are calculated as explained in the above method.

3.3.3 Voltage Magnitude and Angle, VAR Sensitivity Coefficients Calculation Program

Over-voltages, under-voltages, and generator VAR limit violations can be alleviated by adjusting the VAR control variables namely, Switchable shunt capacitors and reactors or SVC, Transformer taps, and Generator voltages. While the voltage angle limit violations can be alleviated by re-scheduling the real power generation in the system.

The above dependent and control variables can be linked if the linearized sensitivity relationships between them are determined. A computer program is developed to determine the sensitivity relationships. This section presents a method of determining the required sensitivity coefficients and the digital solution procedure.

<u>Sensitivity Coefficient Matrix</u>

The calculation methodology of the sensitivity coefficient matrix is given in [28 - 32]. The approach basically utilizes the Jacobian matrix of Newton-Raphson load low program which is in the form of:

$$\begin{bmatrix} \Delta P \\ \Delta Q_L \end{bmatrix} = \begin{pmatrix} \frac{\partial P}{\partial \delta} & \frac{\partial P}{\partial V_L} \\ \frac{\partial Q_L}{\partial \delta} & \frac{\partial Q_L}{\partial V_L} \end{pmatrix} \begin{bmatrix} \Delta \delta \\ \Delta V_L \end{bmatrix} = [J] \begin{bmatrix} \Delta \delta \\ \Delta V_L \end{bmatrix} \tag{3-4}$$

The Jacobian matrix is augmented by equations for power injections at the slack bus and reactive power injections at generator buses with controllable generator voltages and tap settings. For ease of explanation, bus 1 is selected as the slack bus. The sensitivities are in the form of:

$$\begin{bmatrix}\Delta P1\\ \Delta P\\ \Delta Q_L\\ \Delta Qg\end{bmatrix}=\begin{bmatrix}A & B\\ J & D\\ E & F\end{bmatrix}\begin{bmatrix}\Delta\delta\\ \Delta V_L\\ \Delta Vg\\ \Delta T\end{bmatrix} \tag{3-5}$$

where

$$A=\begin{bmatrix}\frac{\partial P1}{\partial\delta} & \frac{\partial P1}{\partial V_L}\end{bmatrix}\quad B=\begin{bmatrix}\frac{\partial P1}{\partial Vg} & \frac{\partial P1}{\partial T}\end{bmatrix}$$

$$D=\begin{bmatrix}\frac{\partial P}{\partial Vg} & \frac{\partial P}{\partial T}\\ \frac{\partial Q_L}{\partial Vg} & \frac{\partial Q_L}{\partial T}\end{bmatrix}$$

$$E=\begin{bmatrix}\frac{\partial Qg}{\partial\delta} & \frac{\partial Qg}{\partial V_L}\end{bmatrix}\quad F=\begin{bmatrix}\frac{\partial Qg}{\partial Vg} & \frac{\partial Qg}{\partial T}\end{bmatrix}$$

By transferring all independent variables (control variables) to the right hand-side and all dependent variables, which are required to be determined, to left-hand side of equation (3-5), the equation can be rewritten as:

$$\begin{bmatrix}\Delta P1\\ \Delta\delta\\ \Delta VL\\ \Delta Qg\end{bmatrix}=\begin{bmatrix}AJ^{-1} & B-AJ^{-1}D\\ J^{-1} & -J^{-1}D\\ EJ^{-1} & F-EJ^{-1}D\end{bmatrix}\begin{bmatrix}\Delta P\\ \Delta QL\\ \Delta Vg\\ \Delta T\end{bmatrix} \tag{3-6}$$

The matrix of equation (3-6) is the sensitivity matrix and contains the desired sensitivity coefficients which give the linear relationship between the dependent variables and control variables. The size of the sensitivity matrix is n x (n + no. of tap changers), where n is the number of buses.

Digital Solution Procedures

An initial load flow solution by NR method is first performed. Using the element of the Jacobian matrix determined by the load flow program at the last iteraction, the vectors A, B, D, E, F of eq. (3-6) are formulated. Next, the elements of the full sensitivity matrix is calculated using these vectors and the inverse of the Jacobian matrix as explained in the above equations.

3.3.4 Variation of Power System Sensitivity Coefficients

The sensitivity coefficients of the power system are calculated to determine the linear relationships between control variables (i.e. var compensator (ΔQ_L), tap changer, (ΔT) generator voltage (ΔVg) and real power generation (ΔP) and dependent variables (i.e. voltage angle ($\Delta\delta$), load bus voltage, var generation (ΔQg) and line power flow(ΔP_{ij}). These coefficients are basically calculated from the Jacobian matrix formulated in the last iteration of the NR load flow solution program. The elements of the Jacobian matrix vary with the changes in the configuration of the network. Hence, adding or deleting an electrical element in the network will change the entire coefficients of the sensitivity matrix.

The sensitivity coefficients are generally assumed to be almost constant as far as the variation of system loading only is concerned. However, this assumption may not be valid in certain situations. For example, the effect of load changes on the sensitivity coefficients were analyzed for IEEE 30-Bus system and IEEE 25-Bus system and the results are shown in Figure 4. Based on these results it can be concluded that the sensitivity coefficients can change significantly with system loadings and the variation can be non-linear especially for heavy system loading.

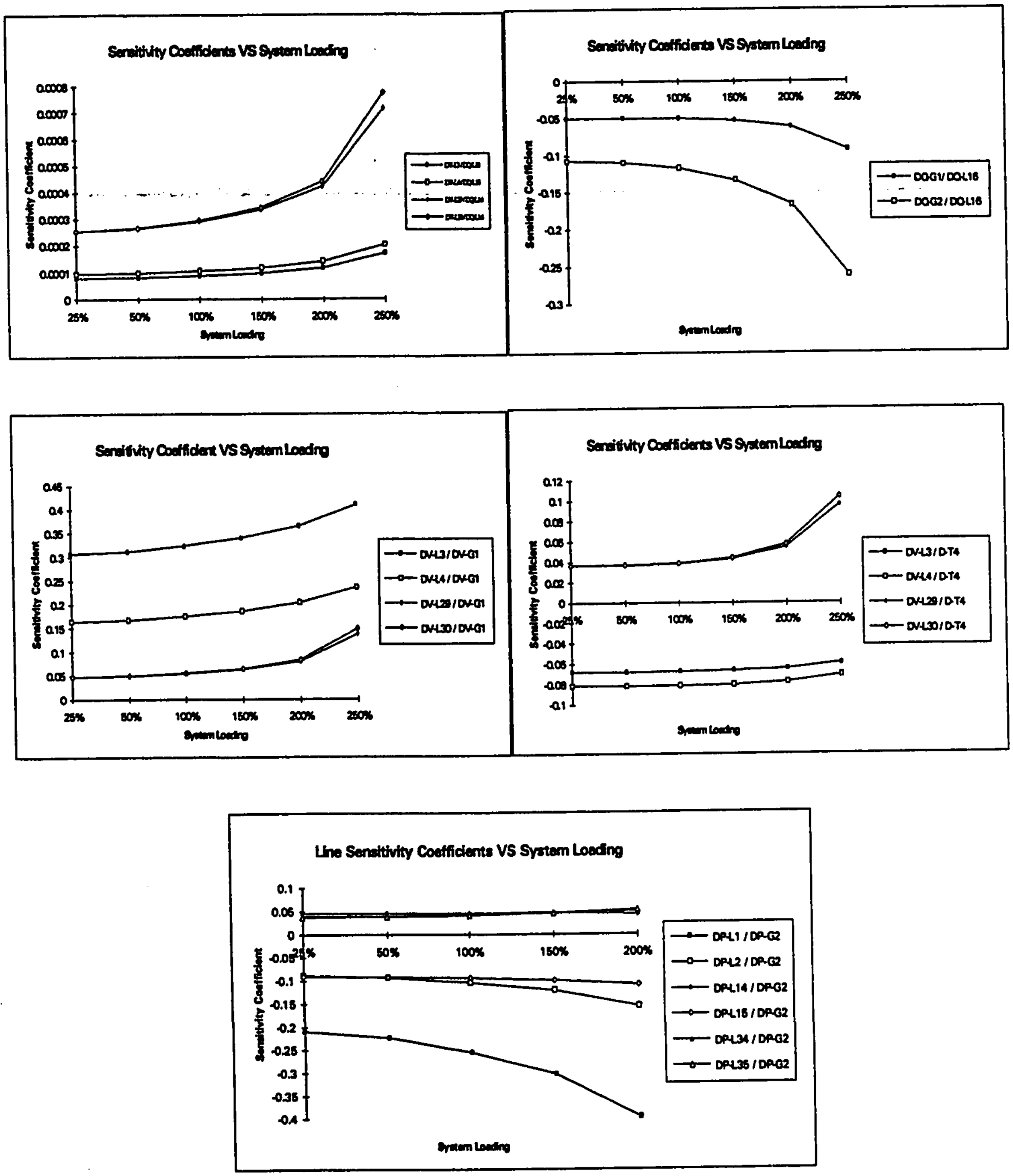

Figure 4: Variation of Sensitivity Coefficients with System Loadings.

3.4 EXPERT SYSTEM FOR LOAD FLOW PLANNING

An expert system is developed to assist in the decision-making of the load flow planning problems which include voltage/reactive power control, line over-loads and voltage angle control. The knowledge required to perform the task is obtained from the literature survey and power dispatch operators of SCECO-East. To alleviate a load flow planning problem, empirical rules are used to generate appropriate control actions. Controls such as shunt capacitors, shunt reactors, transformer tap changers and generator voltages are utilized for voltage/reactive power problem. Line over-loads and voltage angle problems are mitigated by rescheduling the real power generation. In the following, the production rules and inference engine developed to perform the above functions are presented. The basic approach adapted for building the production rules of the developed expert system is based on the utilization of the sensitivity coefficients of the power system discussed in previous sections.

3.4.1 ES for Corrective Generation Re-Scheduling

Knowledge Base

The KB stores an expert's knowledge in the forms of rule, constraint and data. The main data and constraints needed to form the knowledge base are basically the initial power flow on each line and transformer, the initial real power generation of each generator, the upper power flow limits of each line and transformer, the upper and lower limits of real power generation of each generator, and a power flow sensitivity coefficient table for each line or transformer and the generator pair.

Basic Rules

Each rule represents a piece of expert knowledge. The following are the basic rules used for corrective generation re-scheduling, and these rules do not consider any economic dispatch. The rules are basically identified based on the ideas that the rescheduling of the generation for alleviating an overload on a particular line should not cause new overloads and the assigned new real power generations should not exceed the limits of the generators. Also, in case of eliminating of overloads in the first selected line(s) could not be achieved after applying all the possible generation shifting, the process of eliminating of overloads for the remaining overloaded lines should continue.

(Rule 1) If all the lines and transformers carry power within their limits (no overloads),

then exit the program in success.

(Rule 2) If there is (are) overloaded line(s) or transformer(s),

then determine the most overloaded line or transformer.

(Rule 3) If the line or transformer is overloaded,

and the most effective (the highest weighted sensitivity) generator has enough margin to eliminate the overload,

and the desired change in the generation will not cause new overloads,

then change the generation by desired amount.

(Rule 4) If the first selected most effective generator has margin but not enough to eliminate the overload,

and change in the generation to the limit will not cause new overloads,

then change the generation to the limit.

(Rule 5) If the desired change in the generation cause new overloads,

then determine a new change in generation that will not cause new overloads.

(Rule 6) If the generation change in the first selected generator does not eliminate the overload of a line or transformer,

then select the next most effective generator that has generation margin.

(Rule 7) If the first selected line or transformer is still overloaded after applying all possible generation shifts,

then go to the next most overloaded line or transformer.

(Rule 8) If there is still overloaded line or transformer after applying all possible generation shifts,

then print "No Adequate Generation" message and the results.

(Rule 9) If there is no overload after applying the possible generation shifts,

then print the results.

<u>Inference Engine</u>

The inference engine is used to process the rules, facts and constraints given in the knowledge base. It also interacts with I/O interface in order to present the results of conclusions or to get input data. The detailed process of the inference engine is presented as flowchart in Figure 5.

3.4.2 ES for Voltage & VAR Adjustment

In load flow planning the planner should decide the VAR compensation for the

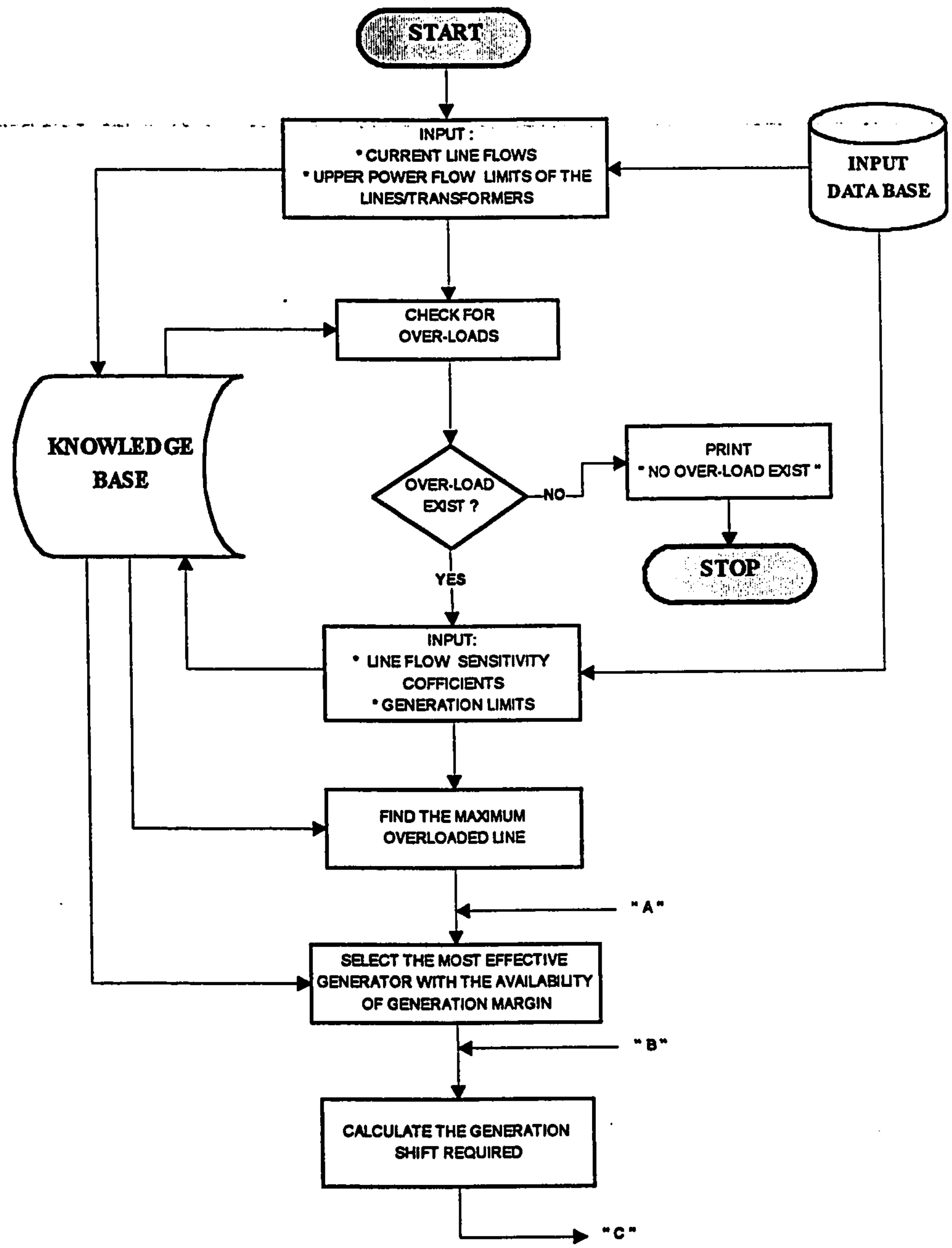

Figure 5: Flow Chart for Generation Shifting Process

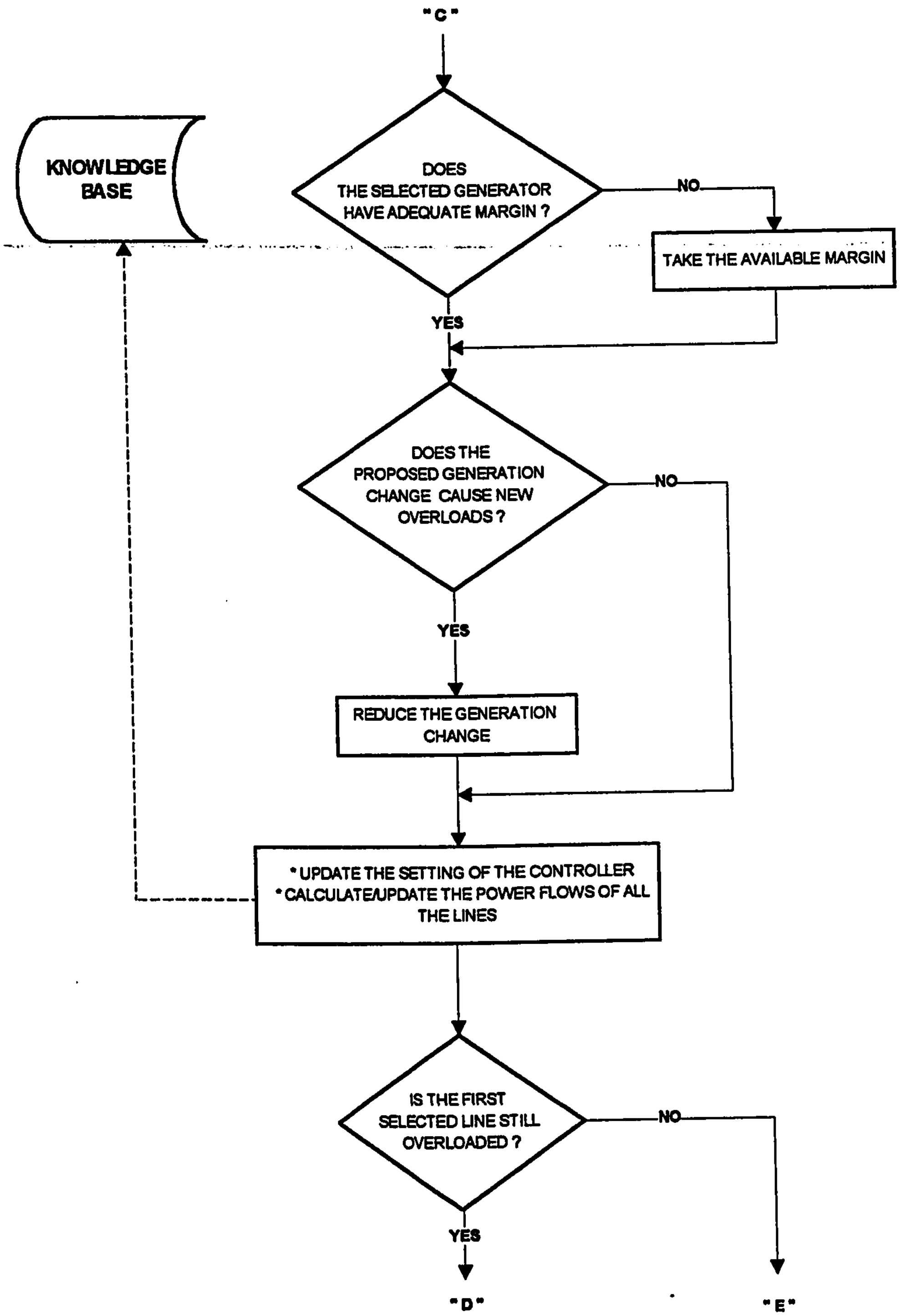

Figure 5: (Continued)

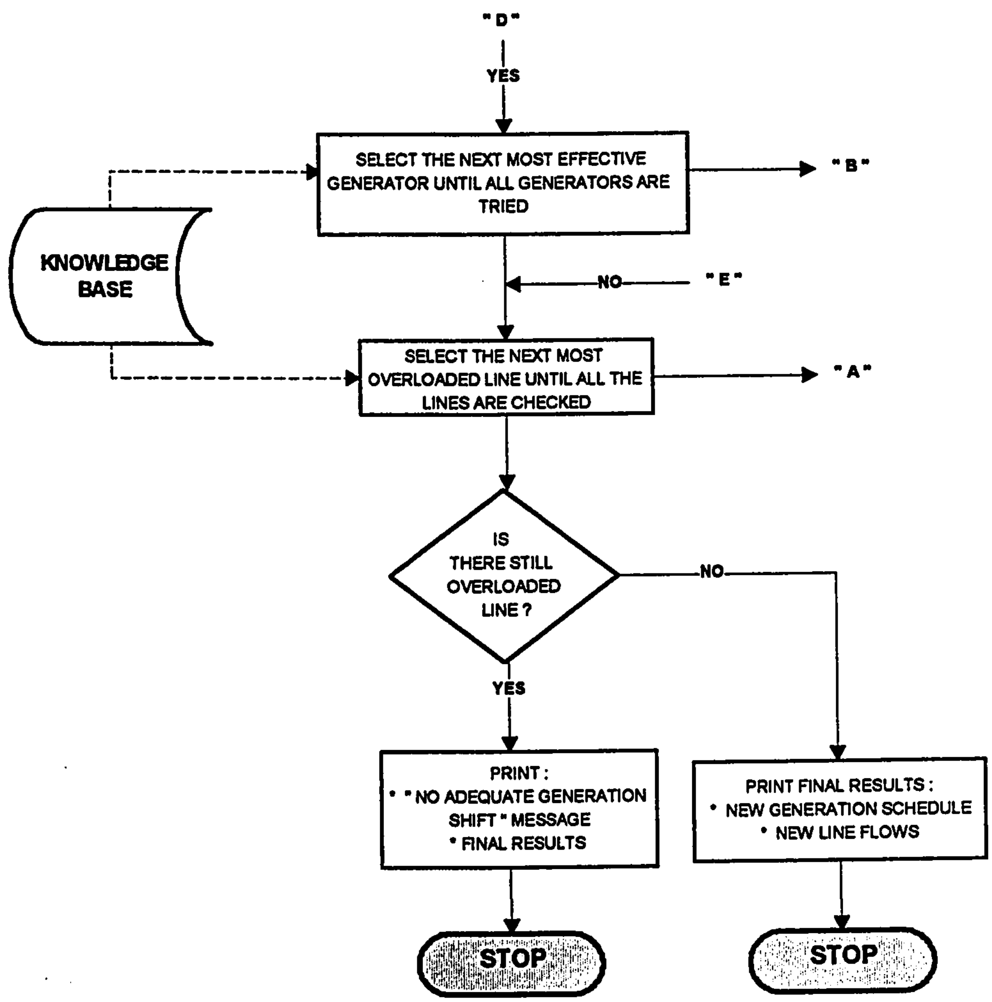

Figure 5: (Continued)

electrical network to retain bus voltages within specific ranges and the VAR generation at the generator buses within their limits. The objective of VAR compensation is to solve over - and under-voltage problems. For example, if a bus voltage violates the normal operating limit, the operator can switch on a shunt reactor or capacitor, raise/lower transformer's tap or adjust the reference voltage of automatic voltage regulator to restore the voltage to a normal value.

Knowledge Base

For the reactive power & voltage control problem, the following data and constraints are needed to form the knowledge base.

1) The initial load flow results.

2) The upper and lower operating limits for load bus voltage.

3) The upper and lower limits of each controller.

4) The upper and lower limits of reactive generation at each generation bus.

5) A sensitivity coefficient table for each load bus voltage and the control measure pair, and for each generator bus reactive power and the control measure pair.

6) The types of available controllers and possible selections (continuous change or discrete change).

   The user shall define the types of the available control elements (i.e. shunt reactor, shunt capacitor, tap changer and generator voltage), and the mode of the adjustment for each type. For example, in case of tap changer or shunt var compensator the user shall define the mode of adjustment as discrete (step by step) and the possible change per step. However, in case of generator voltage or SVC the user shall define the mode of adjustment as continuous.

Empirical Rules

The knowledge base of the developed ES for voltage/VAR control, adapt the operating criteria followed by the power dispatch operators at SCECO-East. Under normal system operation, the voltages of the load buses shall be maintained between 0.95 p.u. and 1.05 p.u. In case of abnormal load bus voltage(s) the operator shall restore the voltage using the available local and nearby (electrically) control elements based on the following priority: 1) Var Compensator; 2) Transformer Tap Changer; 3) Generator Excitation. This priority was set based on economical consideration.

Further to the above criteria, the production rules of the ES are constructed based on utilization of the power system sensitivity coefficients. The sensitivity coefficients are used to identify the nearby (most effective) control elements for the abnormal bus, and the most effective control element is defined by the highest weighted sensitivity. Also, the sensitivity coefficients are needed to calculate the required change in the control measures.

The main ideas which were considered for developing the basic rules are:

1) Restoring the voltage or reactive power at one bus shall not cause new voltage or Var generation problem;

2) If more than one load bus face abonormal voltages, the restoration shall be started with the most under-voltage bus.

The reason of starting with the most under-voltage rather than over-voltage is to avoid the operation of under-voltage relay which will trip all the incomer power lines to that bus.

Based on the above approach, the following empirical rules are identified for voltage/VAR control process:

(Rule 1) If the voltages of all load buses are within the operating limits ($0.95 \leq V \leq 1.05$ p.u.),

and the reactive generations at all generation buses are within their limits,

then exit the program in success.

(Rule 2) If there are more than one load bus with voltage below the lower operating limits,

then start the voltage adjustment with the bus of the highest violation.

(Rule 3) If there are more than one load bus with voltage above the upper operating limit, then start the voltage adjustment with the bus of the highest violation.

(Rule 4) If there is a load bus with voltage below the lower operating limit,

and there is a load bus with voltage above the upper operating limit,

then start the voltage adjustment first with the bus of voltage below the lower operating limit.

(Rule 5) If there is voltage violation,

and there is violation in reactive power generation,

then solve first the voltage violation.

(Rule 6) If a bus voltage is below the lower limit,

then reduce nearby shunt reactor,

or increase nearby shunt capacitor,

or raise the tap of nearby transformer,

or raise the voltage of nearby generator bus.

(Rule 7) If a bus voltage is above the upper limit,

then reduce nearby shunt capacitor,

or increase nearby shunt reactor,

or lower the tap of nearby transformer,

or lower the voltage of nearby generator bus.

(Rule 8) If there are more than one generation bus with the reactive generation violates their lower limits,

then start the VAR adjustment with the bus of the highest violation.

(Rule 9) If there is a generation bus with its VAR generation below the limit,

and there is a generation bus with its VAR generation above the limit.

then start the VAR adjustment first with bus of VAR generation below the limit.

(Rule 10) If a reactive generation is below the limit, then raise the voltage of nearby generator(s).

(Rule 11) If a reactive generation is above the limit,

then lower the voltage of nearby generator(s).

(Rule 12) If there is still voltage or VAR violation after applying all possible control measures,

then print "No Adequate Controllers" message

and print the results.

(Rule 13) If there is no voltage or VAR violation after applying the possible control measures,

then print the results.

The inference engine processes for the above rules is shown in Figure 6.

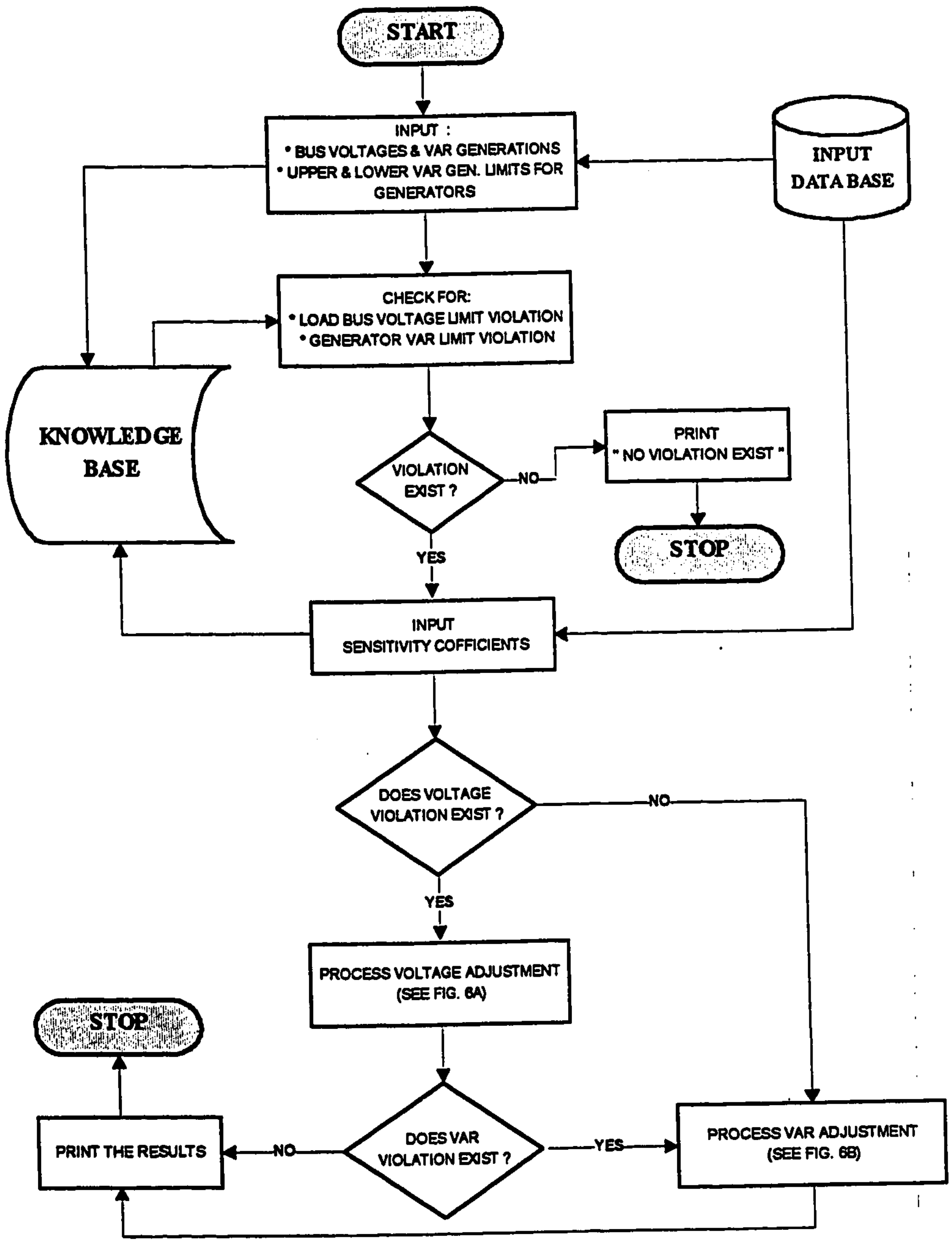

Figure 6: Flow Chart for Voltage & VAR Adjustment Process

123-6

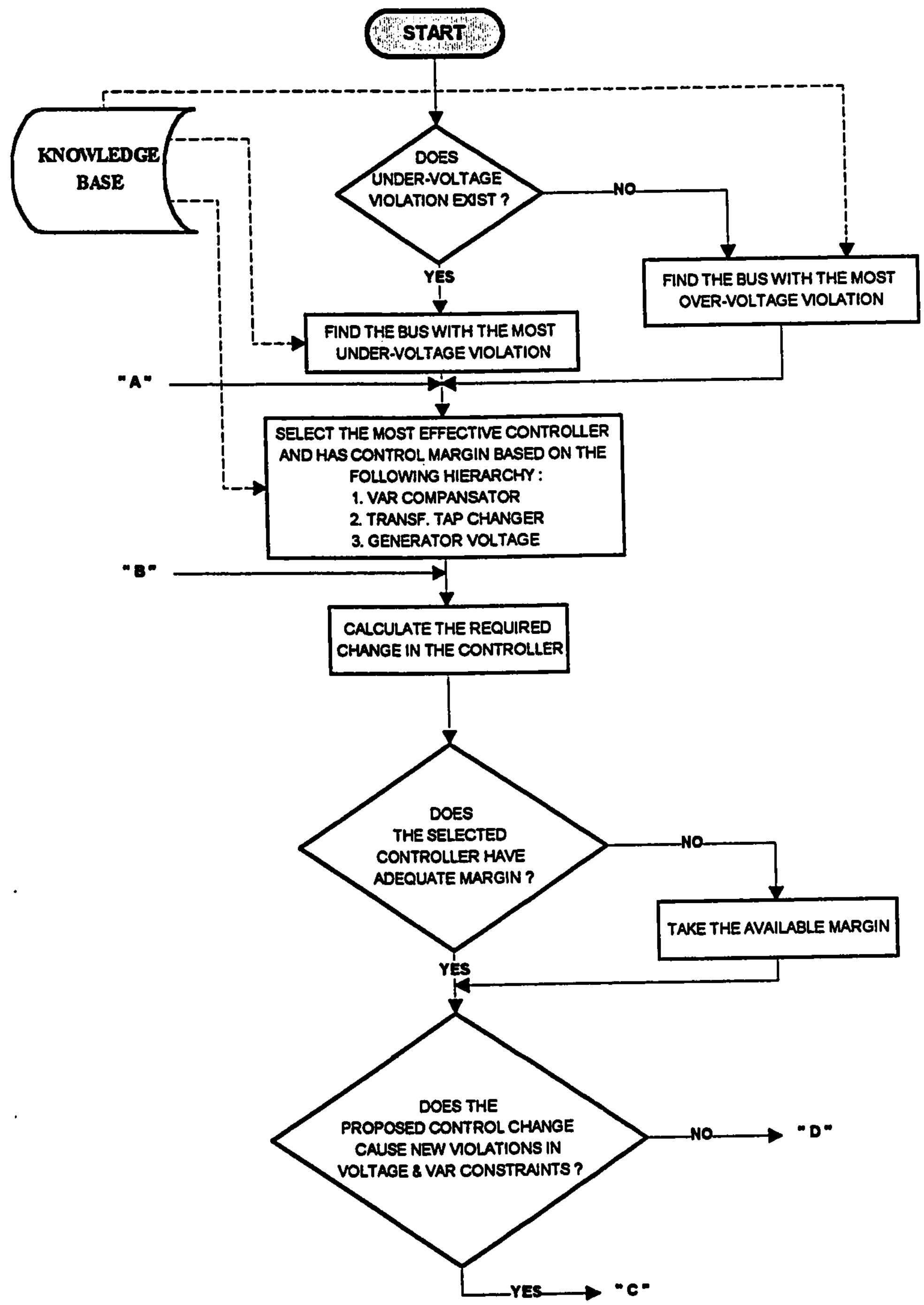

Figure 6A: Flow Chart for Voltage Adjustment Process

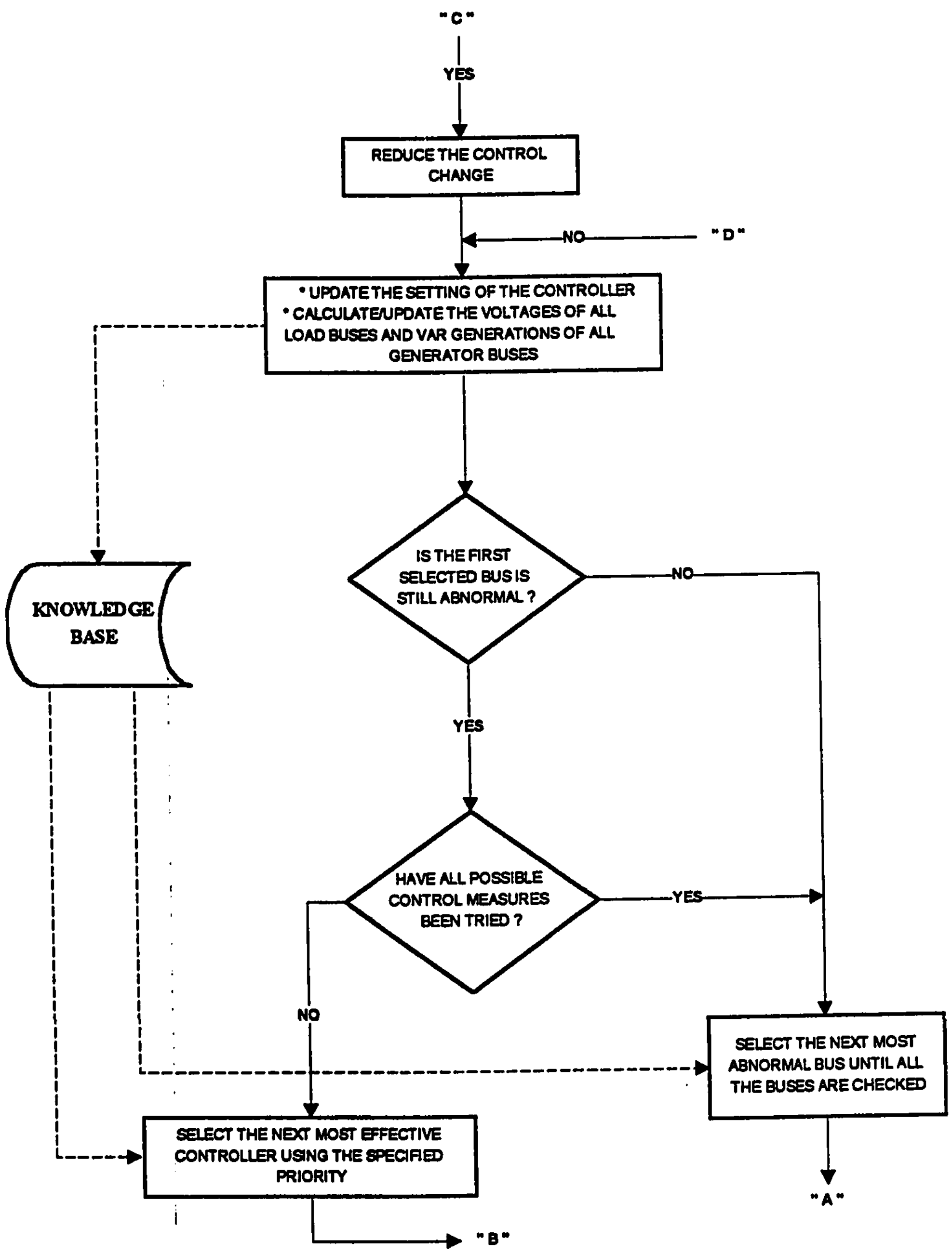

Figure 6A: (Continued)

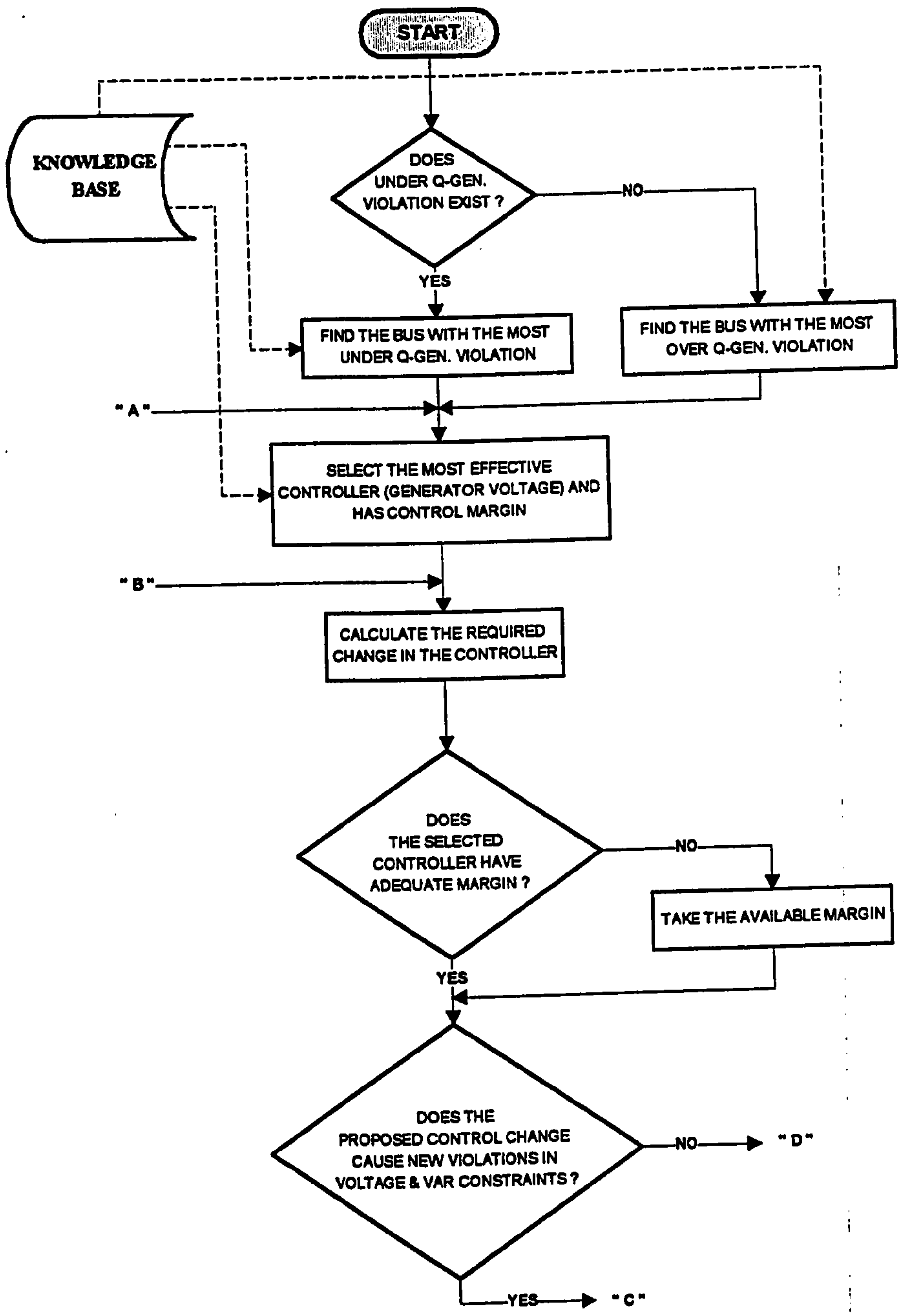

Figure 6B: Flow Chart for VAR Adjustment Process

123-6

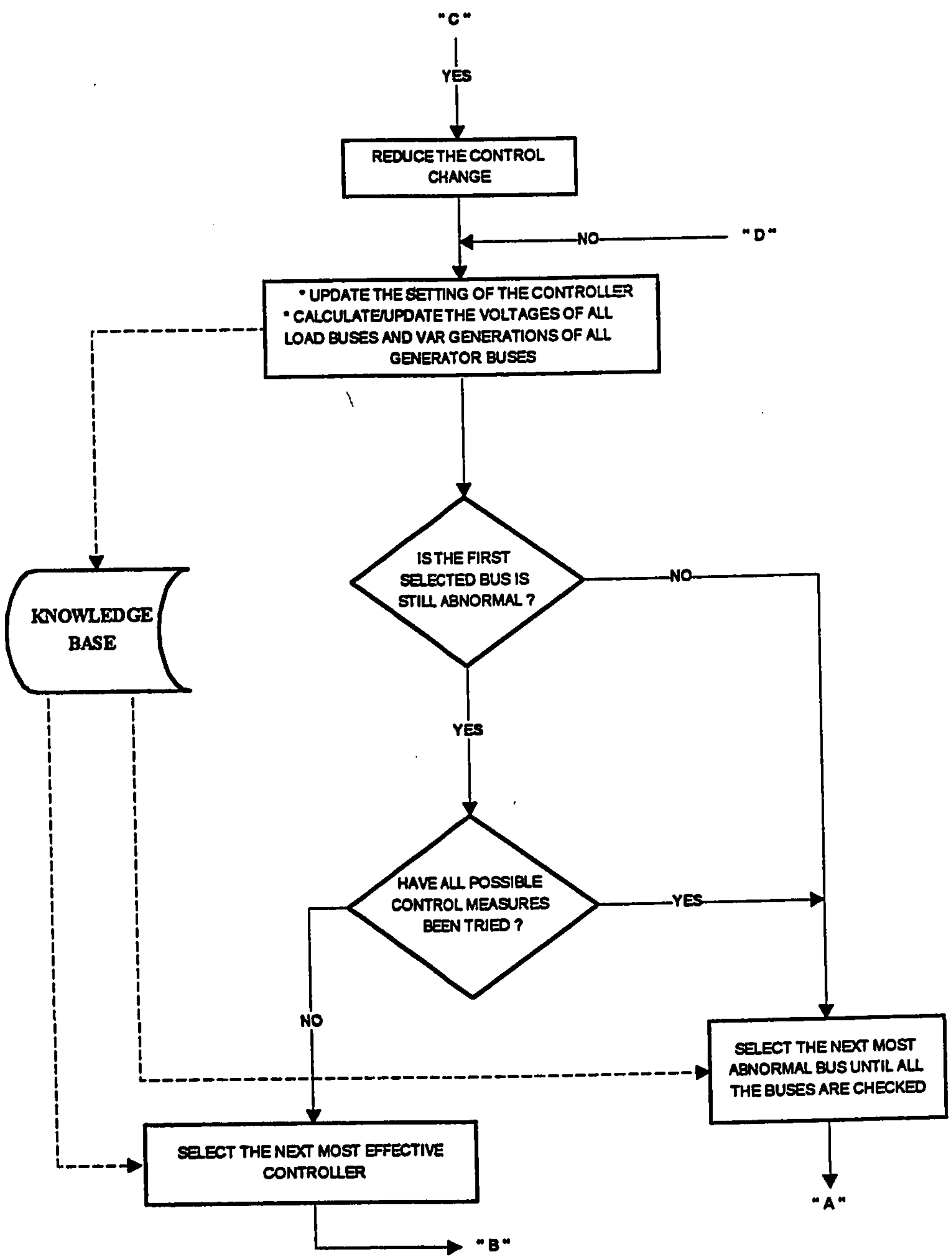

Figure 6B: (Continued)

3.4.3 ES for Voltage Angle Adjustment

Transient stability studies are generally performed by power system planners for determining if the power system will remain in synchronism following major

disturbances. Such studies determine the operating limits, the power angles, for the generators with respect to the swing bus in order to maintain the stability of the system after electrical disturbances.

The developed ES assists the power system operators in decision-making for the control of the voltage angles of the generator buses within the desired limits. The basic idea for controlling the voltage violation is using the linear relation between the voltage angle and real power injection.

Knowledge Base

The KB for voltage angle control problem consists of the following data and constraints:

1) Initial load flow results (voltage angles, real power generation).
2) Desired operating limit for voltage angle with respect to slack bus.
3) Upper and lower real power generation limits of each generator.
4) A voltage angle sensitivity coefficient table for each generator bus and real power pair.

Basic Rules

The basic rules needed for performing the voltage angle control task are given below. The rules are basically identified based on the idea that the rescheduling of the generation for adjusting the voltage angle at a particular generation bus should not cause new voltage angle violations and the assigned new real power generations should not exceed the limits of the generators.

(Rule 1) If all the generators operate within acceptable power angle limits,

then exit the program in success.

(Rule 2) If there is (are) abnormal generator buses,

then determine the bus with the most violation in power angle limit.

(Rule 3) If the bus has abnormal power angle,

and the most effective generator has enough margin to eliminate the violation,

and the desired change in the generation will not cause new violations,

then change the generation by the desired amount.

(Rule 4) If the first selected most effective generator has margin but not enough to eliminate the power angle violation, and change in the generation to the limit will not cause new violation,

then change the generation to the limit.

(Rule 5) If the desired change in the generation cause new violations,

then determine a new change in generation that will not cause new violations.

(Rule 6) If the generation change in the first selected generator does not eliminate the power angle violation of the selected bus,

then go to next most effective generator that has generation margin.

(Rule 7) If the first selected generator bus still has power angle violation after applying all possible generation shifts,

then go to the next generation bus.

(Rule 8) If there is still power angle violation after applying all possible generation shifts, then print a message "No Adequate Controller", and the results.

(Rule 9) If there is no any more power angle violation after generation shifts,

then print the results.

Inference Engine

The solution process using the above knowledge base for voltage angle violation problem is given in Figure 7.

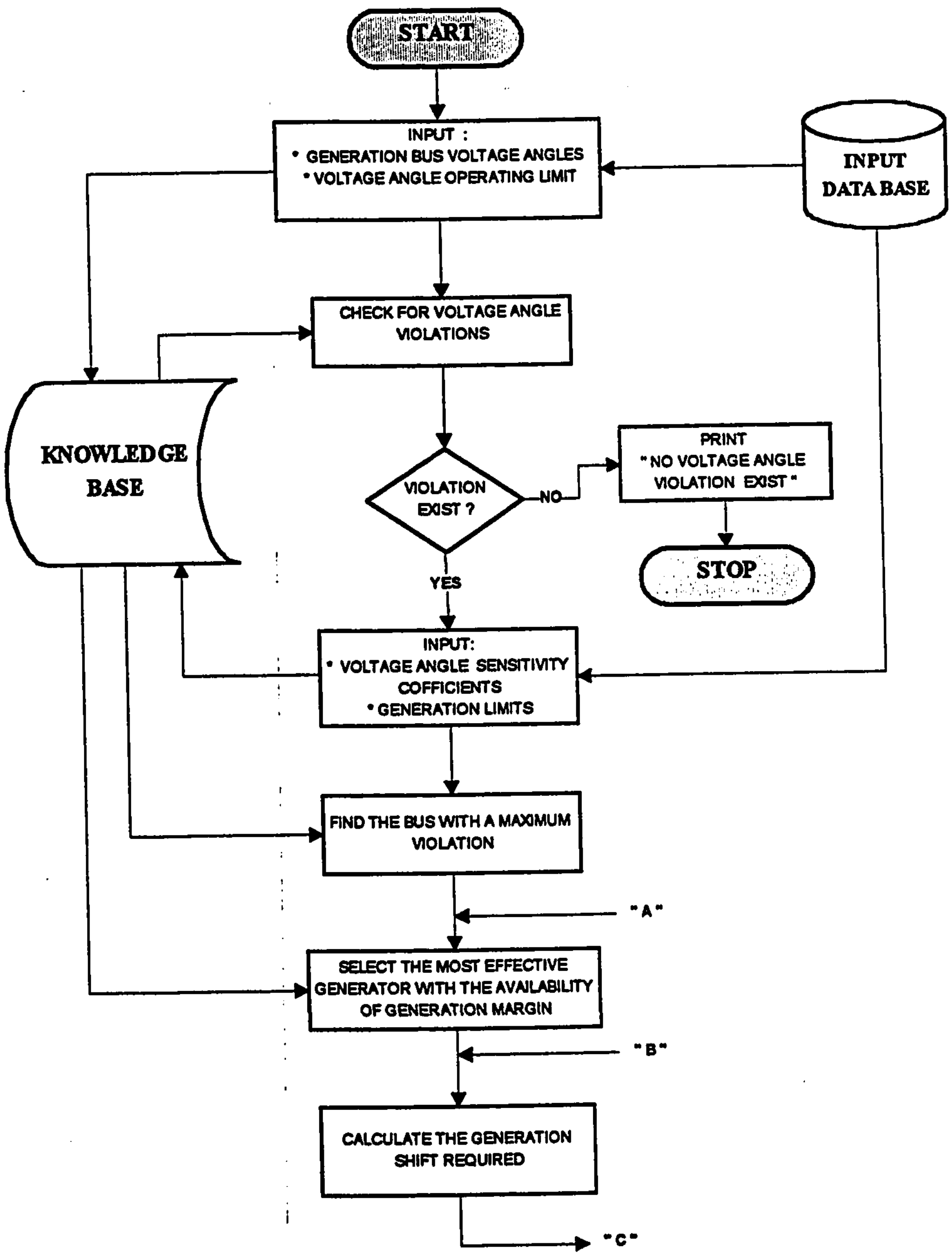

Figure 7: Flow Chart for Voltage Angle Adjustment Process

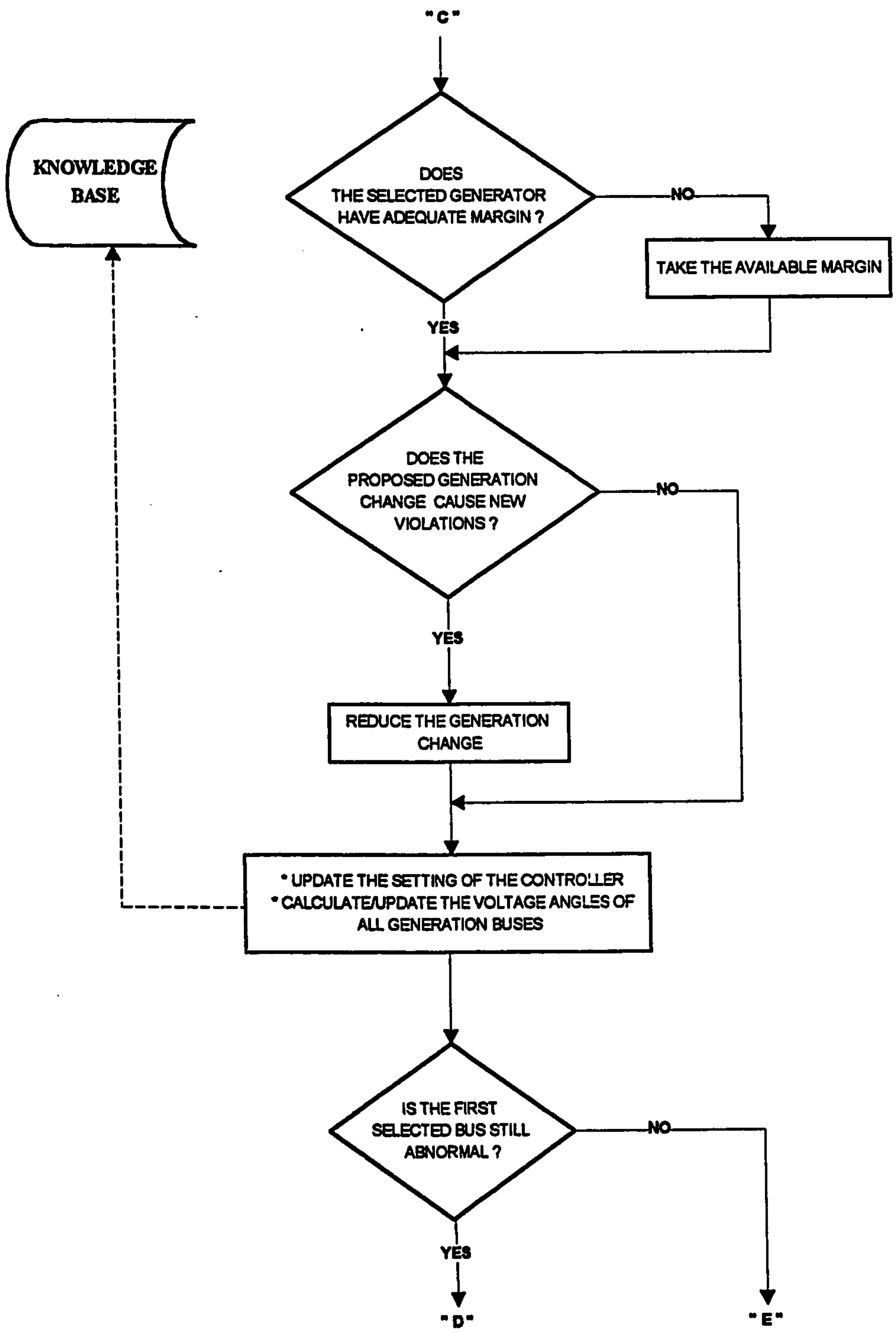

Figure 7: (Continued)

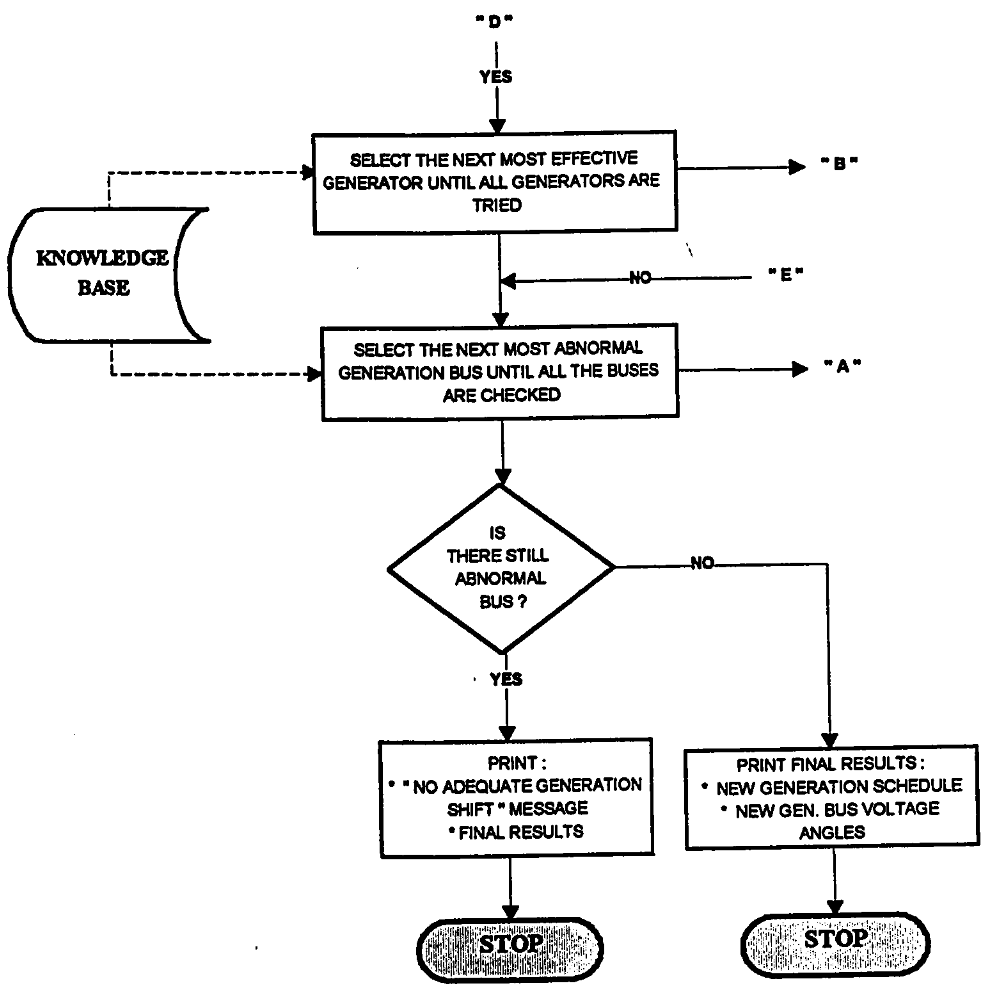

Figure 7: (Continued)

CHAPTER 4

CASE STUDIES AND RESULTS

4.1 INTRODUCTION

The developed ES prototype has been tested on several different power systems using IBM compatible PC with 486 Intel processor. It does not have any limitation as far as the application is concerned. It can be applied to any power system, not specific to a particular power system configuration.

This Chapter will present the test cases performed and the results obtained. First, the test cases and results for the voltage and reactive power control on the modified IEEE 14 and 30 bus power systems will be presented. Next, the test cases and results for the corrective generation re-scheduling on the modified IEEE 6 and 25 bus power systems will be shown. Finally, the chapter will present the test cases and results for voltage angle adjustment on the modified IEEE 25 bus power system.

4.2 TEST CASES AND RESULTS FOR VOLTAGE AND VAR CONTROL

CASE 1: MODIFIED IEEE 14-BUS SYSTEM

In this case a modified IEEE 14-bus system, Fig. 8, is chosen to test the capability of the developed ES prototype. The system parameters are taken from reference [33]. However, the following changes were made to test the capability of the ES with the application of various control elements and to cause voltage and Var limit violations.

1) A shunt reactor has been added at bus 14,

2) A shunt capacitor has been added at bus 13,

3) Loads have been changed at buses 4, 9 and 13,

4) Starting voltages have been changed at buses 1, 2, 3, 6 and 8,

5) Tap settings of transformers have been slightly changed.

The full line and bus data of the modified IEEE 14-bus system are given in Tables 2 and 3, respectively. Table 4 presents the limits on the control and dependent variables, the initial load flow results and the final results calculated by the ES prototype, and the results produced by the load flow program to verify the calculated results.

The load flow study was performed for the base case system state and the results are given in Table 4 as "initial state". A review of this system state, along with the limits of the dependent variables (Table 4), indicates the following abnormal conditions:

Under-Voltages: Bus # 13

Over-Voltages: Bus # 7 and 9

Low reactive power: Bus # 2

High reactive power: Bus # 3 and 8

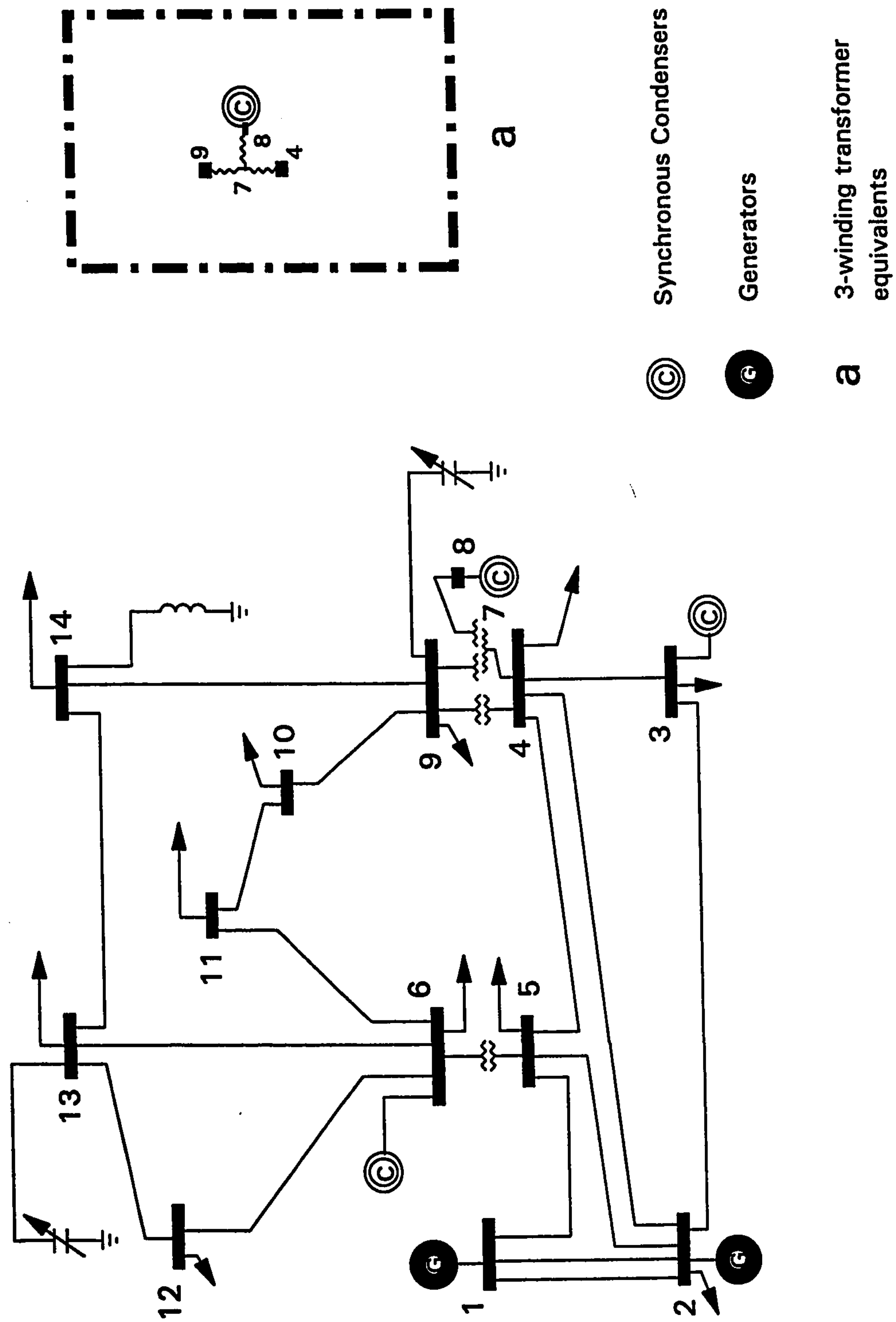

Figure 8: Single Line Diagram for Modified IEEE 14-Bus System

The developed ES prototype was used to alleviate the abnormal conditions. The new system conditions are described in Table 4 as "final state (1)".

The load-flow study was repeated after applying the suggested changes in the control variables by the ES to verify the accuracy of the prototype. The new final results of the load flow program are given in Table 4 under column "Final State (2)".

A comparison of the initial and final system states indicates no further violations to the limits.

The following example of manual calculation using sensitivity coefficient is given for illustration and to relate the result obtained here with the ES output.

Per the ES output (Table 4) the control variables Q_9, Q_{13}, V_1, V_2, V_3 and V_8 were changed from their initial values by Δ equals; -19, 9.0, 0.0356, 0.0319, -0.0548 and -0.0191, respectively.

For example, the new voltage at the load bus no. 7 can be calculated as follow:

$$V7\ (new) = V7\ (init.) + \Delta V7$$

$$= 1.058 + \Delta V7$$

$$\Delta V7 = [S] \begin{bmatrix} \Delta Q9 \\ \Delta Q13 \\ \Delta V1 \\ \Delta V2 \\ \Delta V3 \\ \Delta V8 \end{bmatrix} = [0.000526\ \ 0.000088\ \ 0.0554\ \ 0.1761\ \ 0.1043\ \ 0.4626] \begin{bmatrix} -19.0 \\ 9.0 \\ 0.0356 \\ 0.0319 \\ -0.0548 \\ -0.0191 \end{bmatrix}$$

Where [S] is a vector of sensitivity coefficients

Therefore,

$$V7\ (new) = 1.058 + (-0.01551) = 1.042$$

And this value is the same as that obtained by ES (See Table 4).

TABLE 2: Line Data of the IEEE 14-Bus System

Line Designation	Resistance p.u.*	Reactance p.u.*	Line Charging p.u.*
1-2	0.01938	0.05917	0.0264
1-5	0.05403	0.22304	0.0246
2-3	0.04699	0.19797	0.0219
2-4	0.05811	0.17632	0.0187
2-5	0.05695	0.17388	0.0170
3-4	0.06701	0.17103	0.0173
4-5	0.01335	0.04211	0.0064
4-7	0.0	0.20912	0.0
4-9	0.0	0.55618	0.0
5-6	0.0	0.25202	0.0
6-11	0.09498	0.19890	0.0
6-12	0.12291	0.25581	0.0
6-13	0.06615	0.13027	0.0
7-8	0.0	0.17615	0.0
7-9	0.0	0.11001	0.0
9-10	0.03181	0.08450	0.0
9-14	0.12711	0.27038	0.0
10-11	0.08205	0.19207	0.0
12-13	0.22092	0.19988	0.0
13-14	0.17093	0.34802	0.0

* Impedance and line-charging susceptance in p.u. on 100 MVA base. Line charging one-half of total charging of line.

TABLE 3: Bus Data of the modified IEEE 14-Bus System

Bus number	Bus type	Starting bus voltage: Magnitude p.u	Starting bus voltage: Phase angle p.u.	Generation: MW	Generation: M Var	Load: MW	Load: M Var
1	SLK	1.05	0.0	0.0	0.0	0.0	0.0
2	PV	1.025	0.0	40.0	0.0	21.7	12.7
3	PV	1.08	0.0	0.0	0.0	94.2	19.0
4	PQ	1.0	0.0	0.0	0.0	37.8	-3.9
5	PQ	1.0	0.0	0.0	0.0	7.6	1.6
6	PV	1.0	0.0	0.0	0.0	11.2	7.5
7	PQ	1.0	0.0	0.0	0.0	0.0	0.0
8	PV	1.1	0.0	0.0	0.0	0.0	0.0
9	PQ	1.0	0.0	0.0	19.0	10.5	-6.6
10	PQ	1.0	0.0	0.0	0.0	9.0	5.8
11	PQ	1.0	0.0	0.0	0.0	3.5	1.8
12	PQ	1.0	0.0	0.0	0.0	6.1	1.6
13	PQ	1.0	0.	0.0	0.0	43.5	35.8
14	PQ	1.0	0.0	0.0	0.0	14.9	5.0

TABLE 4: Limits on the Variables and the Results of the Study on the Modified IEEE 14-Bus System.

Variables	Limits Lower	Limits Upper	Variation Mode	Variation Steps	Initial State	Final State (1)	Final State (2)
<u>Control Variables</u>							
VAR Sources (MVAR)							
Q_9	0.0	19.0	D	1	19.0	0.0	0.0
Q_{13}	0.0	9.0	D	1	0.0	9.0	9.0
Q_{14}	0.0	-9.0 *	D	1	0.0	0.0	0.0
Transformer Taps							
$t_{4\text{-}7}$	0.9	1.1	D	17	0.975	0.975	0.975
$t_{9\text{-}4}$	0.9	1.1	D	17	0.9625	0.9625	0.9625
$t_{5\text{-}6}$	0.9	1.1	D	17	0.9375	0.9375	0.9375
Generator Voltages (p.u)							
V_1	1.0	1.1	C		1.05	1.0856	1.0856
V_2	1.0	1.1	C		1.025	1.0569	1.0569
V_3	1.0	1.1	C		1.08	1.0252	1.0252
V_6	1.0	1.1	C		1.0	1.0	1.0
V_8	1.0	1.1	C		1.1	1.0809	1.0809
<u>Dependent Variables</u>							
Voltages of Load Buses (p.u.)							
V_4	0.95	1.05			1.039	1.038	1.038
V_5	0.95	1.05			1.035	1.044	1.044
V_7	0.95	1.05			1.058	1.042	1.041
V_9	0.95	1.05			1.057	1.034	1.032
V_{10}	0.95	1.05			1.040	1.021	1.019
V_{11}	0.95	1.05			1.017	1.008	1.007
V_{12}	0.95	1.05			0.969	0.972	0.971
V_{13}	0.95	1.05			0.946	0.952	0.950
V_{14}	0.95	1.05			0.990	0.979	0.977
Generator Reactive Power (MVAR)							
Q_1	-40.0	40.0			-11.649	8.413	8.848
Q_2	-40.0	50.0			-61.879	0.536	3.834
Q_3	0.0	40.0			103.702	17.305	22.051
Q_6	-6.0	50.0			47.542	43.468	45.132
Q_8	-6.0	24.0			26.255	24.0	24.676

D = Discrete, C = Continuous,
(1) = Results of ES, (2) = Results of Load Flow Program
* = The negative sign means a shunt reactor.

123-6

CASE 2: **MODIFIED IEEE 30-BUS SYSTEM**

The ES prototype was also tested on the modified IEEE 30-bus system. The single line diagram of the system is given in Figure 9. The system parameters are taken from [33], however, the bus data and control variables have been modified as indicated below:

1) Shunt reactors added at buses 16 & 28,

2) Shunt capacitors added at buses 10, 15 & 24,

3) Tap settings of transformer have been slightly changed.

4) Maximum Var capbilities have been changed at generator buses 8, 11 & 13.

The initial system state, determined by the load-flow program, and presented in Table 5 indicates the following abnormalities:

Under-Voltages : Bus # 14, 15, 18, 19, 20, 23, 24, 25, 26, 27, 29 & 30

High reactive power: Bus # 2, 8, 11 & 13.

The final state of the 30-bus system is given in Table 5. As shown in the table the developed ES prototype has alleviated all the abnormal voltages and reactive powers.

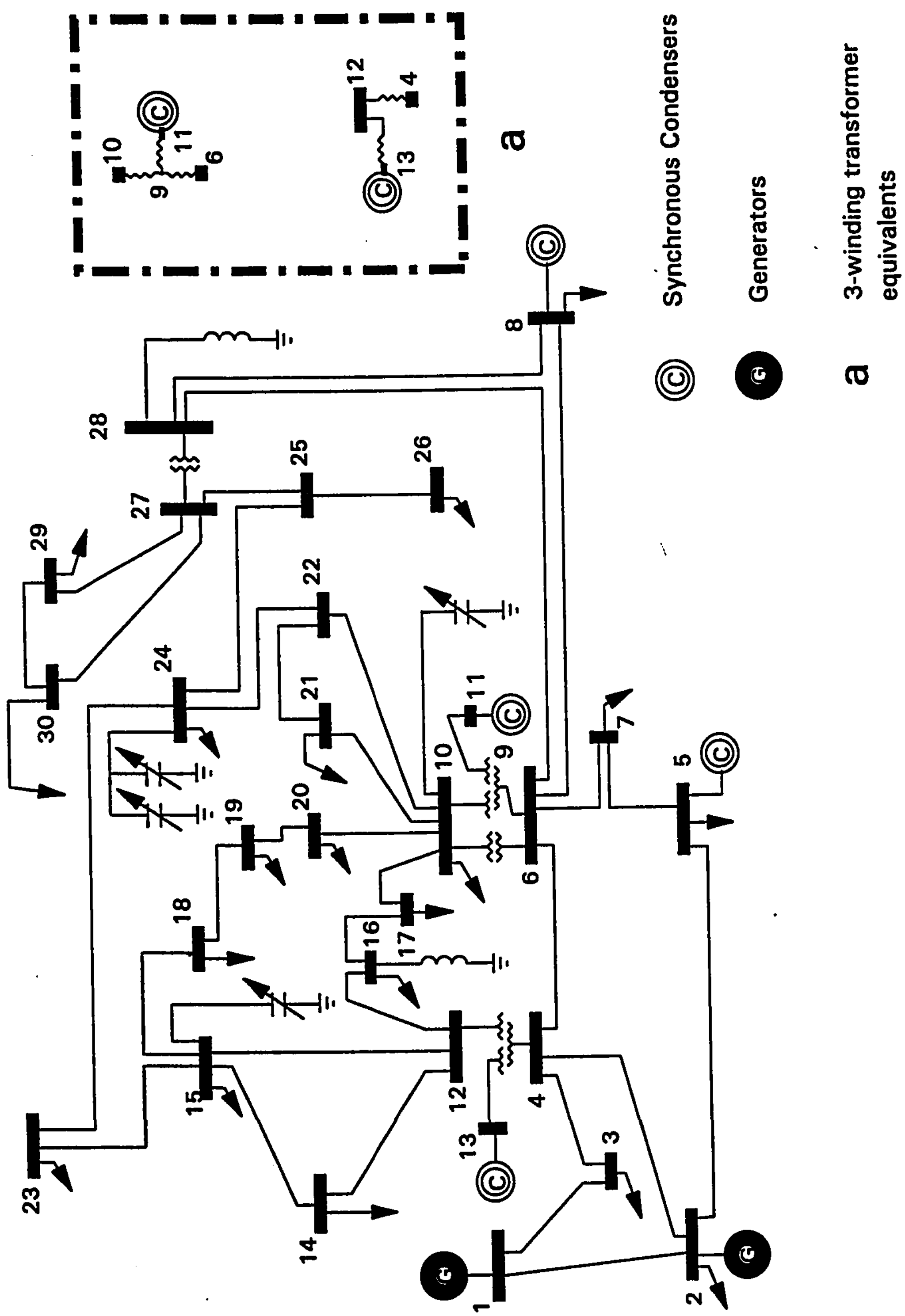

Figure 9: Single Line Diagram for Modified IEEE 30-Bus System

TABLE 5: Limits on the Variables and the Results of the Study on the Modified IEEE 30-Bus System.

Variables		Limits		Variation		Initial	Final	Final
		Lower	Upper	Mode	Steps	State	State (1)	State (2)
Control Variables								
VAR Sources (MVAR)	Q_{10}	0.0	19.0	D	1	19.0	19.0	19.0
	Q_{15}	0.0	25.0	D	1	0.0	25.0	25.0
	Q_{16}	0.0	-10.0*	D	1	0.0	0.0	0.0
	Q_{24}	0.0	2 x 4.3	D	2	4.3	8.6	8.6
	Q_{28}	0.0	-6.0*	D	1	0.0	0.0	0.0
Transformer Taps	t_{4-12}	0.9	1.1	D	17	0.9250	0.9375	0.9375
	t_{6-9}	0.9	1.1	D	17	0.9750	1.0125	1.0125
	t_{6-10}	0.9	1.1	D	17	0.9625	0.9625	0.9625
	t_{28-27}	0.9	1.1	D	17	0.9625	0.9625	0.9625
Generator Voltages (p.u)	V_1	1.0	1.1	C		1.060	1.0942	1.0942
	V_2	1.0	1.1	C		1.045	1.0621	1.0621
	V_5	1.0	1.1	C		1.01	1.0308	1.0308
	V_8	1.0	1.1	C		1.01	1.0287	1.0287
	V_{11}	1.0	1.1	C		1.082	1.1	1.1
	V_{13}	1.0	1.1	C		1.071	1.0953	1.0953
Dependent Variables								
Voltages of Load Buses (p.u.)	V_3	0.95	1.05			1.018	1.046	1.045
	V_4	0.95	1.05			1.010	1.036	1.035
	V_6	0.95	1.05			1.004	1.027	1.026
	V_7	0.95	1.05			0.999	1.021	1.020
	V_9	0.95	1.05			1.001	1.048	1.047
	V_{10}	0.95	1.05			0.975	1.029	1.027
	V_{12}	0.95	1.05			0.980	1.032	1.030
	V_{14}	0.95	1.05			0.948	1.014	1.010
	V_{15}	0.95	1.05			0.921	0.999	0.995
	V_{16}	0.95	1.05			0.971	1.024	1.022
	V_{17}	0.95	1.05			0.968	1.022	1.020
	V_{18}	0.95	1.05			0.895	0.969	0.965
	V_{19}	0.95	1.05			0.884	0.954	0.950
	V_{20}	0.95	1.05			0.900	0.966	0.963
	V_{21}	0.95	1.05			0.958	1.014	1.012
	V_{22}	0.95	1.05			0.957	1.014	1.012
	V_{23}	0.95	1.05			0.900	0.975	0.971
	V_{24}	0.95	1.05			0.924	0.989	0.986
	V_{25}	0.95	1.05			0.930	0.981	0.978
	V_{26}	0.95	1.05			0.911	0.962	0.960
	V_{27}	0.95	1.05			0.944	0.984	0.983
	V_{28}	0.95	1.05			1.001	1.024	1.024
	V_{29}	0.95	1.05			0.922	0.964	0.962
	V_{30}	0.95	1.05			0.910	0.952	0.950

Table 5 (Cont'd.)

Generator								
Reactive	Q_1	-40.0	40.0			-32.779	1.591	3.034
Power	Q_2	-40.0	50.0			76.664	32.355	32.722
(MVAR)	Q_5	-40.0	40.0			39.084	39.319	39.638
	Q_8	-10.0	50.0			54.567	43.112	44.513
	Q_{11}	-6.0	35.0			41.919	27.796	28.075
	Q_{13}	-6.0	50.0			69.618	50.0	51.225

D = Discrete, C = Continuous,
(1) = Results obtained from ES,
(2) = Results of Load Flow Programs
* The negative sign means a shunt reactor.

4.3 TEST CASES/RESULTS FOR CORRECTIVE GENERATION RE-SCHEDULING

CASE 1: IEEE 6-BUS SYSTEM

The developed ES prototype was tested on the IEEE 6-bus system whose single line diagram is shown in Figure 10 [34]. The initial operating state of the system as obtained from the load flow simulation is given in Table 6. By comparing the initial state of the load flows with the upper limits of the lines, the following overloads can be identified:

Overloaded Lines: 1-4, 1-5 & 3-6.

The 4th column of Table 5 presents the final state of the power flow after modifying the generation schedules by the ES prototype. The modified generation schedules, are given in the last column of Table 7. The new results clearly indicate no further overloads.

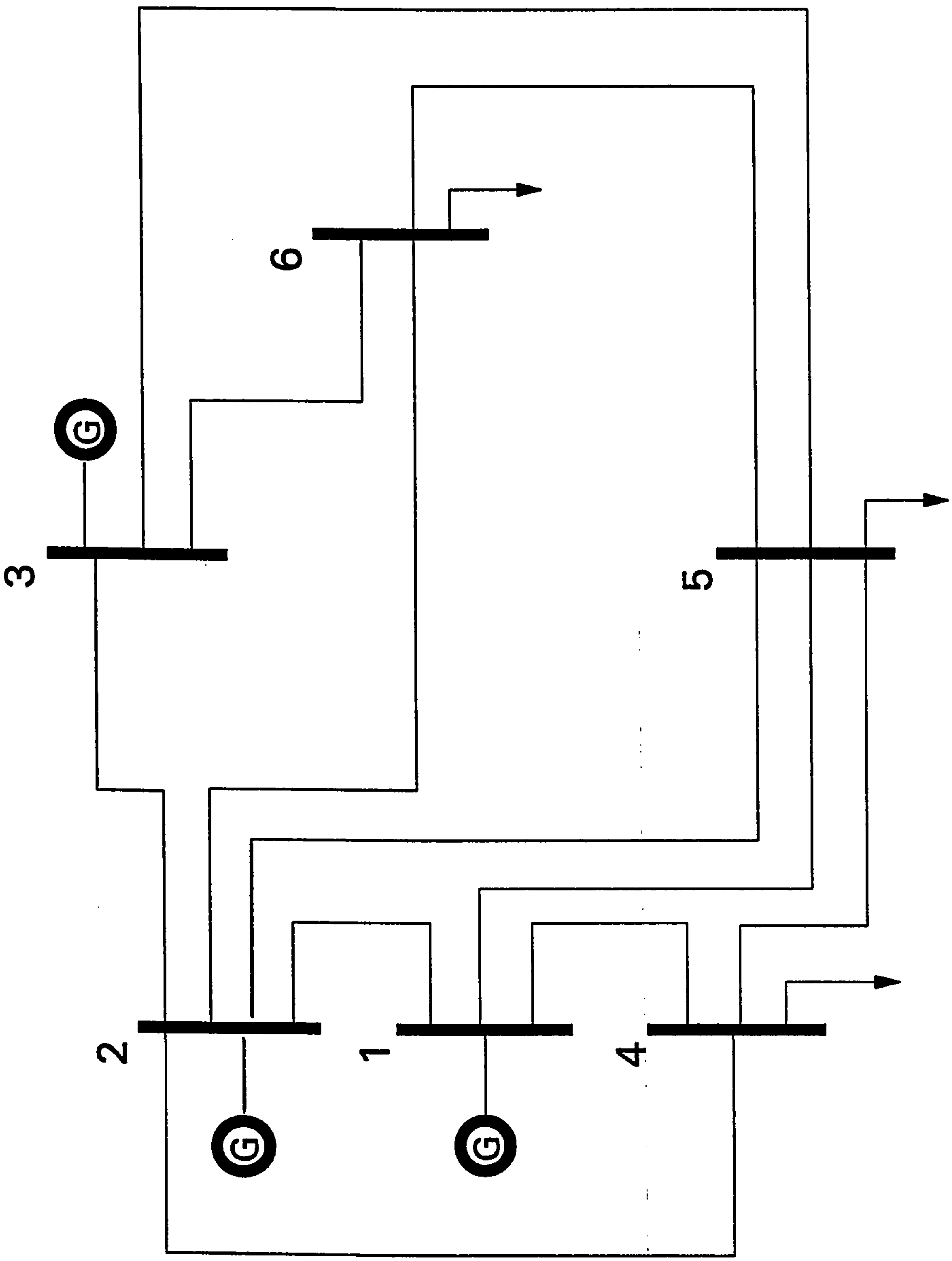

Figure 10: Single Line Diagram for IEEE 6-Bus System

TABLE 6: Initial and Final Power Flows For IEEE 6-Bus System

Line designation	Upper limit (MW)	Initial state (MW)	Final state[1] (MW)	Final state[2] (MW)
1-2	33.5	28.687	4.9217	5.495
1-4	36.0	43.583*	27.0423	27.378
1-5	30.0	35.599*	25.0961	25.313
2-3	15.5	2.930	10.0545	10.062
2-4	60.0	33.091	49.4528	49.423
2-5	30.0	15.514	21.9075	21.892
2-6	40.0	26.248	32.3859	32.392
3-5	23.0	19.117	20.4684	20.453
3-6	40.0	43.773*	40.0	40.002
4-5	30.0	4.083	4.2485	4.250
5-6	15.0	1.614	-0.6303	-0.597

* = Overload
1 = Results obtained from ES
2 = Results of Load Flow Program

TABLE 7: Basic and Modified Generation Schedules for 6-Bus System

Generation designation	Limit (MW)		Initial state (MW)	Final state (MW)
	Lower	Upper		
G_2	37.5	150.0	50.0	108.3078
G_3	45.0	180.0	60.0	50.4864

CASE 2: **IEEE 25-BUS SYSTEM**

The developed prototype was applied for alleviating overloads on the IEEE 25-bus system. [35]. The single line diagram of the system is shown in Figure 11.

Table 8 presents the initial power flows and limits of the lines. Based on the data given in Table 8, the following overloaded lines can be found:

The Overloaded Lines were:- 2-7, 3-14, 4-19, 5-17, 8-9, 17-18, 20-21, & 22-24.

The modified generation schedules as suggested by the ES prototype are given in Table 9. Using the modified generation schedules the new power flows are given in Table 8 and the results indicate that all the overloaded lines were alleviated.

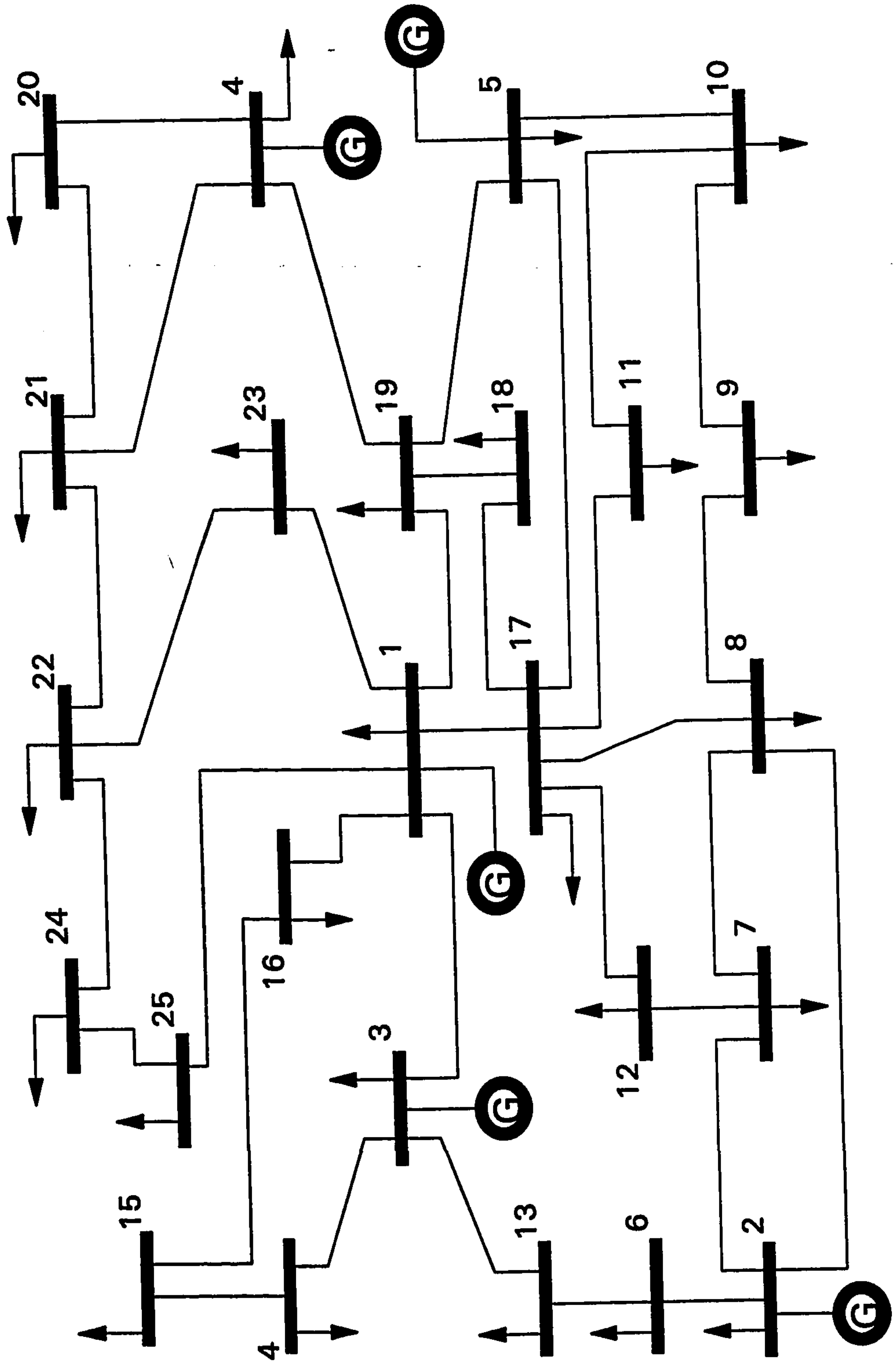

Figure 11: Single Line Diagram for IEEE 25-Bus System

TABLE 8: Initial and Final Power Flows For IEEE 25-Bus System

Line designation	Upper limit (MW)	Initial state (MW)	Final state[1] (MW)	Final state[2] (MW)
1-3	100.0	-38.744	-29.9775	-29.732
1-16	120.0	40.802	44.3736	44.483
1-17	60.0	-16.161	5.0406	5.724
1-19	50.0	0.115	10.6586	11.013
1-23	40.0	26.699	30.0786	30.201
1-25	90.0	33.006	36.0076	36.097
2-6	100.0	22.740	9.2942	9.306
2-7	30.0	33.006*	27.9505	27.943
2-8	50.0	33.404	27.4219	27.412
3-13	40.0	17.807	31.0880	31.361
3-14	40.0	42.204*	38.2654	38.186
4-19	40.0	-43.671*	-37.3873	-37.300
4-20	40.0	39.305	36.2128	36.170
4-21	40.0	24.373	21.1814	21.134
5-10	55.0	31.752	30.0423	30.069
5-17	75.0	83.496*	69.7797	69.717
5-19	85.0	59.768	45.2163	45.249
6-13	25.0	7.415	-5.6876	-5.792
7-8	15.0	5.972	4.69	4.684
7-12	15.0	12.324	7.89	7.872
8-9	6.0	6.564*	6.0	5.988
8-17	10.0	7.171	0.7029	0.675
9-10	10.0	-8.457	-9.0198	-9.032
10-11	15.0	7.741	5.5136	5.521
11-17	20.0	2.680	0.4874	0.491
12-17	12.0	2.268	-2.1272	-2.151
14-15	25.0	20.124	16.5683	16.474
15-16	15.0	-9.994	-13.4987	-13.599
17-18	15.0	18.008*	13.7353	13.706
18-19	9.0	2.745	-1.4002	-1.445
20-21	12.0	13.703*	10.6985	10.649
21-22	25.0	17.378	11.3708	11.257
22-23	15.0	-10.671	-13.7011	-13.766
22-24	7.0	7.924*	5.0319	4.965
24-25	20.0	-7.131	-0.9723	-10.048

* = Overload
1 = Results obtained from ES
2 = Results of Load Flow Program

TABLE 9: Basic and Modified Generation Schedules for IEEE 25-Bus System

Generation designation	Limit (MW)		Initial state (MW)	Final state (MW)
	Lower	Upper		
G_2	70.00	150.00	100.00	74.6583
G_3	50.00	200.00	150.00	150.00
G_4	30.00	60.00	50.00	50.00
G_5	100.00	250.00	200.00	170.0282

4.4 TEST CASES AND RESULTS FOR VOLTAGE ANGLE ADJUSTMENT

CASE: IEEE 25-BUS SYSTEM

The developed ES prototype has been tested for voltage angle violation problem on the IEEE 25-bus system shown in Figure 12. The system contains five generators. The initial voltage angles of the generator buses as obtained from the load flow study are presented in Table 10.

The case study desires to maintain all the voltage angle difference, between the slack bus and other generator buses, within 4.0 degrees. Therefore, the voltage angles of the following generator buses must be adjusted.

Generator Buses: Bus # 2, 3 & 5.

Table 11 presents the modified generation schedules and according to the new schedules the new voltage angles were calculated and presented in Table 10.

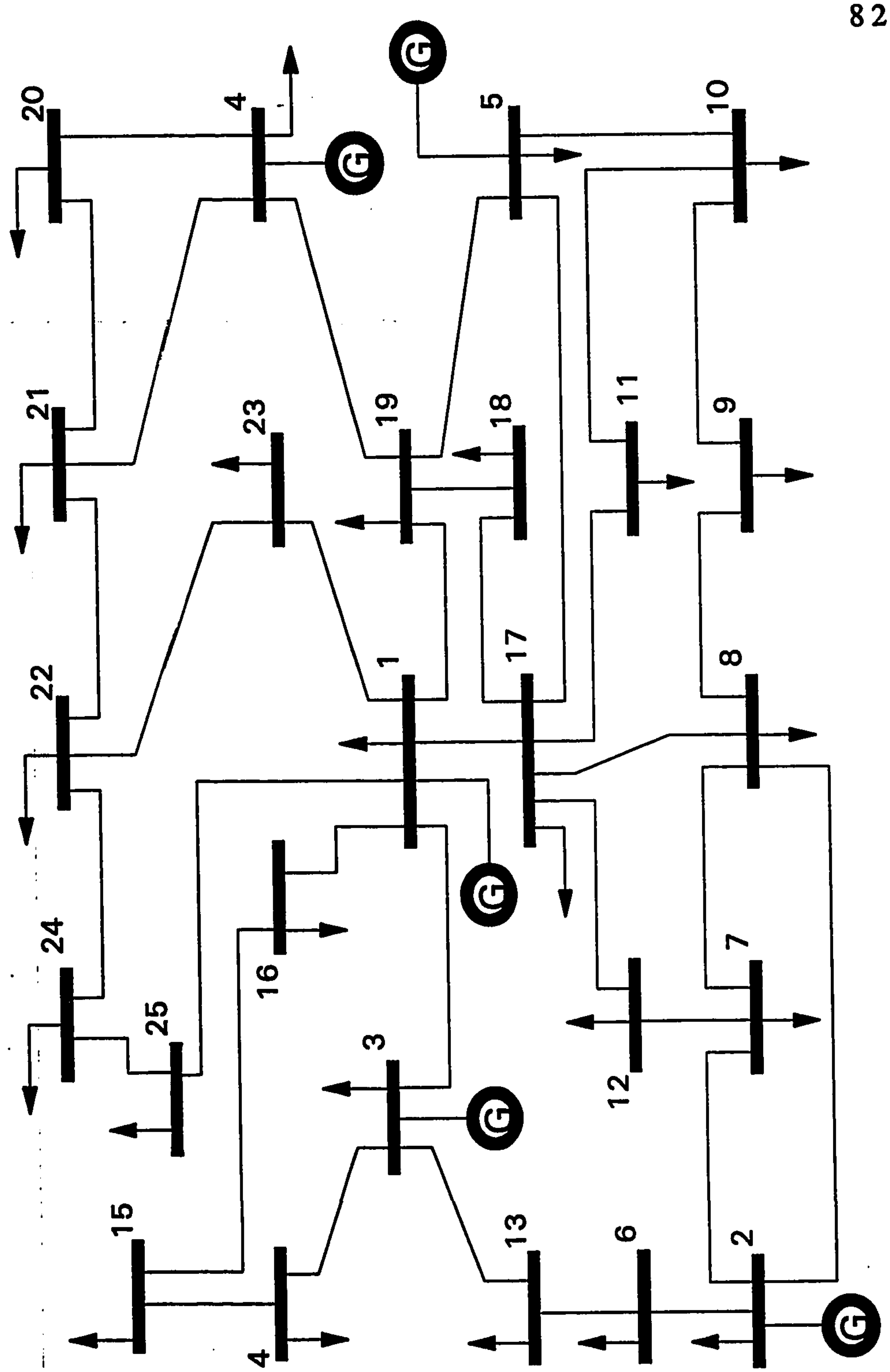

Figure 12: Single Line Diagram of IEEE 25-Bus System for Voltage Angle Adjustment

TABLE 10: Initial and Final Voltage Angles of Generator Buses for IEEE 25 Bus System.

Generation bus designation	Initial voltage angle (deg.)	Final voltage Angle[1] (deg.)	Final voltage Angle[2] (deg.)
1 SLK	0.0	0.0	0.0
2	9.461*	3.7222	3.642
3	6.851*	4.00	3.962
4	-1.271	-3.3083	-3.4
5	9.242*	3.3513	3.259

* = Voltage Angle Violation
1 = Results obtained from ES
2 = Results of Load Flow Program

TABLE 11: Basic and Modified Generation Schedules for IEEE 25-Bus System

Generation designation	Limit (MW)		Initial state (MW)	Final state (MW)
	Lower	Upper		
G_2	70.0	150.0	100.0	82.7664
G_3	50.0	200.0	150.0	134.7947
G_4	30.0	60.0	50.0	60.0
G_5	100.0	250.0	200.0	153.0287

CHAPTER 5

CONCLUSIONS AND RECOMMENDATIONS

5.1 SUMMARY

An expert system prototype for load flow planning has been developed. It has the following main features.

1) Windows input/output interface facilities to give an interactive user-friendly environment.

2) The expert system prototype can detect overloads in the power system and propose new generation schedule to alleviate the overloads.

3) The prototype can detect over-voltages, under-voltages, and violations of reactive power limits of generators and it provides remedial measures to alleviate these violations.

4) The expert system can control the voltage angles of the generation buses within the desired limits.

The expert system has been developed using an expert system programming tool called CLIPS and adopting a technique which is based on the use of sensitivity factors.

The developed expert system has been tested on various examples of power systems. Test results show very satisfactory results.

In addition to the expert system a power system application program has been developed using FORTRAN language to calculate the load flow and sensitivity factors of the power system. The application program has been integrated with the expert system in a single package.

5.2 CONCLUSION

Expert system methodology provides the power system engineers with new powerful tools. Intelligent load flow engine is a good example for demonstrating the possibility of expert system. It assists the planners in the load flow planning and the operators in decision-making for power system control. The reasoning of expert system seems to be more acceptable to the power system planners and operators than the solution of linear programing or other analytical procedures, because the former matches human heuristic thought process.

5.3 CONTRIBUTION OF THE THESIS

The thesis mainly contributes in the power system engineering by presenting an intelligent load flow planning tool to the power system engineers. The developed expert system can be useful at least in the following aspects:

1) It can be used to assist in decision-making for abnormal operating conditions in the power system.

2) It can be used to train power system operators.

3) It can be used to assist in system planning.

5.4 RECOMMENDATIONS FOR FUTURE WORK

Expert system being part of the family of artificial intelligence still constitute a young technology. So far most of the expert system in the power system applications are prototypes and under field tests.

There are several applications in power system operation in which the expert system can be applied. The following areas can be considered for future works:

1) Voltage and var control under voltage instability condition.

2) Economic generation shifting with line flow constraints.

APPENDICES

APPENDIX - A

COMPUTER LISTING OF POWER SYSTEM APPLICATION PROGRAMS

LOAD FLOW & SENSITIVITY COEFFICIENTS CALCULATION PROGRAM

```
C########################################################################
C
C      THIS PROGRAM PERFORMS THE FOLLOWINGS:-
C
C       1. LOAD FLOW STUDY BY NEWTON-RAPHSON METHOD .
C
C       2. CALCULATES THE SENSITIVITY COEFFICIENTS FOR POWER SYSTEM
C
C       3. CALCULATES THE SENSITIVITY COEFFICIENTS FOR REAL POWER FLOW
C
C########################################################################
C
C    THE FORMAT OF THE INPUT DATA SHALL BE AS FOLLOW:-
C
C########################################################################
C
C
C
C NO.OF BUSES(N), NO.OF LINES(C)  ,'REFERENCE BUS'('00')
C
C KVOLT(BASE),MVA(BASE)
C
C LINE,'1ST BUS','2ND BUS', IMPD.(R,X),TOT. SUSP.(G,YC), P. FLOW LIM.
C    =       =      =               =            =              =
C    =       =      =               =            =              =
C
C NOTE:- 1. THE LINE PARAMETERS SHALL BE IN PER-UNIT.
C          2. THE LINE SUSP. IS THE TOTAL.
C
C NUMBER OF STATIC VAR COMPANSATORS(NSC)
C 'SVC LOCATION',PRES. SETTING(+ OR - MVAR),MAX. LIM.(+ OR - MVAR),
C MIN. LIM. (+ OR - MVAR),CHANG. PER SETTING(+ MVAR)
C    =            =              =                 =
C NOTE:- '+' FOR CAPACITOR , '-' FOR REACTOR
C
C NUMBER OF TAP CHANGERS
C 'FIRST BUS','SECOND BUS',TAP POSITION,'TAP LOCATION',MAX. LIM.,
C MIN. LIM.,CHANG. PER SETTING
C    =            =               =                =
C
C POWER MISMATCH,'SLACK BUS',SLACK VOLT.(REAL,IMAG),LOAD(REAL,IMAG),
C MAX. VOLTAGE(P.U.),MIN. VOLTAGE(P.U.),Q.G.MAX,Q.G.MIN
C
C NO.OF VOLTAGE CONT. BUSES,NO. OF GENERATORS,NO. OF CONDENSERS
C 'BUS NAME',REAL POWER GENERATED,BUS VOLT.(REAL,IMAG),LOAD(REAL,IMAG),
```

```
C MAX. VOLTAGE(P.U.),MIN. VOLTAGE(P.U.),P.G.MAX,P.G.MIN,Q.G.MAX,Q.G.MIN
C      =          =                 =
C      =          =                 =
C
C NOTE: FIRST ENTERS THE DATA OF ALL GENERATORS
C
C NO.OF LOAD BUSES
C 'BUS NAME',LOAD(REAL,IMAG),GEN.(REAL,IMAG),INIT.BUS VOLT(REAL,IMAG)'
C P.G.MAX,P.G.MIN
C      =          =                 =
C      =          =                 =
C
C
C
C##################################################################
C
C    MAIN PROGRAM:
C                       READS DATA FOR LOAD FLOW STUDY BY NEWTON-RAPHSON
C
C##################################################################
C
      CHARACTER*10 REF,BUSA,BUSB,SLACK,B(30),AA,A(30),LAB(30),BB
      CHARACTER*10 TA(10),TB(10),TLOC,BUSSC(20),LINE(50,2)
      COMPLEX Z,YB(30,30),VS,S(30),V(30),YBN,YP(30,30),YC
      COMPLEX YPN,SLL,GL(30),GR(30)
      REAL P(30),TAP(10),QSC(20),MAXLIM(20),MINLIM(20),DSETT(20)
      REAL ERROR,KVB,MVAB,TMAX(10),TMIN(10),DTAP(10),MXVOLT(10)
      REAL MNVOLT(10),IMAG,PGMAX(30),PGMIN(30),FLIM(50)
      REAL QGMAX(20),QGMIN(20)
      INTEGER N,C,NLBUS,NVBUS,E,NTC,NSC,LN,NG,NC
      COMMON /MAIN0/V,B,S,LAB,P,SLL,GR,GL,NTC,TA,TB,TAP
      COMMON /MAIN1/YB
      COMMON /MAIN2/YP
      COMMON /MAIN3/NSC,BUSSC,QSC,MAXLIM,MINLIM,DSETT
      COMMON /MAIN7/MXVOLT,MNVOLT,TMAX,TMIN,DTAP,LINE,PGMAX,PGMIN,
     * FLIM,C,QGMAX,QGMIN
C
      OPEN(5,FILE='LFLOW.DAT',STATUS='OLD')
      OPEN(6,FILE='LFLOW.OUT')
C
       K=0
      READ(5,*)N,C,REF
C
      READ(5,*)KVB,MVAB
C
       DO 15 I=1,C
         E=I
         READ(5,*)LN,BUSA,BUSB,Z,YC,FLIM(LN)
         LINE(LN,1)=BUSA
         LINE(LN,2)=BUSB
         CALL YBUS(A,K,C,N,REF,BUSA,BUSB,Z,E,YC)
15     CONTINUE
```

```
C
      READ(5,*)NSC
      IF(NSC.NE.0)THEN
      DO 11 I=1,NSC
      READ(5,*)BUSSC(I),QSC(I),MAXLIM(I),MINLIM(I),DSETT(I)
       QSC(I)=QSC(I)/MVAB
       MAXLIM(I)=MAXLIM(I)/MVAB
       MINLIM(I)=MINLIM(I)/MVAB
11    CONTINUE
       DO 12 I=1,NSC
       DO 13 J=1,N
       IF(BUSSC(I).EQ.A(J))THEN
       IMAG=AIMAG(YB(J,J))+QSC(I)
       YB(J,J)=CMPLX(REAL(YB(J,J)),IMAG)
       GO TO 12
       ENDIF
13     CONTINUE
12     CONTINUE
      ENDIF
C
      READ(5,*)NTC
      IF(NTC.NE.0)THEN
      DO 1 I=1,NTC
      READ(5,*)TA(I),TB(I),TAP(I),TLOC,TMAX(I),TMIN(I),DTAP(I)
      IF(TLOC.NE.TA(I))THEN
      TB(I)=TA(I)
      TA(I)=TLOC
      ENDIF
1     CONTINUE
      ENDIF
C
       READ(5,*)ERROR,SLACK,VS,SLL,MXVOLT(1),MNVOLT(1),QGMAX(1)
     * ,QGMIN(1)
         B(1)=SLACK
         LAB(1)='SLK'
         V(1)=VS
         V(1)=V(1)/KVB
         GL(1)=SLL
         GR(1)=(0.0,0.0)
         SLL=SLL/MVAB
       READ(5,*)NVBUS,NG,NC
        IF(NVBUS.NE.0)THEN
        DO 10 I=2,NVBUS+1
       READ(5,*)B(I),P(I),V(I),GL(I),MXVOLT(I),MNVOLT(I),PGMAX(I),
              *PGMIN(I),QGMAX(I),QGMIN(I)
          LAB(I)='PV'
          P(I)=P(I)/MVAB
          V(I)=V(I)/KVB
          GL(I)=GL(I)/MVAB
         GR(I)=(0.0,0.0)
10     CONTINUE
        ENDIF
```

```
      READ(5,*)NLBUS
      DO 20 I=NVBUS+2,N
         READ(5,*)B(I),S(I),GR(I),V(I),PGMAX(I),PGMIN(I)
         LAB(I)='PQ'
         GL(I)=S(I)
         S(I)=-S(I)/MVAB
         S(I)=S(I)+GR(I)/MVAB
         V(I)=V(I)/KVB
20    CONTINUE
      DO 30 I=1,N
       DO 40 J=1,N
      IF(B(I).EQ.A(J))THEN
       DO 50 K=1,N
          YBN=YB(J,K)
          YPN=YP(J,K)
          YB(J,K)=YB(I,K)
          YP(J,K)=YP(I,K)
          YB(I,K)=YBN
          YP(I,K)=YPN
50      CONTINUE
C
        DO 60 K=1,N
           YBN=YB(K,J)
           YPN=YP(K,J)
           YB(K,J)=YB(K,I)
           YP(K,J)=YP(K,I)
           YB(K,I)=YBN
           YP(K,I)=YPN
60        CONTINUE
C          GO TO 1000
            BB=A(J)
            A(J)=A(I)
            A(I)=BB
            GO TO 30
          ENDIF
40      CONTINUE
30      CONTINUE
C
        CLOSE(5)
C
        CALL LFLOW(N,NLBUS,NVBUS,ERROR,KVB,MVAB,NG,NC)
C
        CLOSE(6)
C
         STOP
         END
```

```
C
C************************************************************************
C
C     SUBROUTINE OF YBUS:
C                                  DETERMINES THE Y-BUS MATRIX.
C
C************************************************************************
C
        SUBROUTINE YBUS(A,K,C,N,REF,LINA,LINB,Z,E,YC)
        CHARACTER *10 A(30),LINA,LINB,REF
        INTEGER M,D,PP,Q,R,W,Y,C,N,E
        COMPLEX YB(30,30),L(30,30),Z,SUM,YC,YP(30,30)
C
        COMMON /MAIN1/YB
        COMMON /MAIN2/YP
C
        IF (K .EQ. 0) THEN
        DO 100 I=1,N
        DO 110 J=1,N
        L(I,J)=(0.0,0.0)
        YP(I,J)=(0.0,0.0)
 110    CONTINUE
 100    CONTINUE
        A(1)=LINA
        A(2)=LINB
        L(1,2)=1/Z
        YP(1,2)=YC/2.0
        L(2,1)=L(1,2)
        YP(2,1)=YP(1,2)
        K=2
        RETURN
        ELSE
        PP=0
        Q=0
        ENDIF
        DO 20 I=1,K
        IF(LINA.EQ.A(I))THEN
        M=I
        PP=1
        GO TO 40
        ENDIF
  20    CONTINUE
        K=K+1
        M=K
        A(M)=LINA
  40    DO 30 I=1,K
        IF(LINB.EQ.A(I))THEN
        D=I
        Q=1
        GO TO 45
        ENDIF
  30    CONTINUE
```

```
      K=K+1
      D=K
      A(D)=LINB
45    J=PP*Q
      IF(J.EQ.1)THEN
      L(M,D)=L(M,D)+1/Z
      YP(M,D)=YP(M,D)+YC/2.0
      L(D,M)=L(M,D)
      YP(D,M)=YP(M,D)
      ELSE
      L(M,D)=1/Z
      YP(M,D)=YC/2.0
      L(D,M)=L(M,D)
      YP(D,M)=YC/2.0
      ENDIF
      IF(E.EQ.C)THEN
      DO 60 I=1,N
      DO 70 J=1,N
      IF(I.EQ.J)THEN
      SUM=(0.0,0.0)
      DO 90 R=1,N
      IF(R.NE.I)THEN
C
      SUM=SUM+L(I,R)+YP(I,R)
C
      ENDIF
90    CONTINUE
      YB(I,J)=SUM
      ELSE
C
      YB(I,J)=-1*L(I,J)
C
      ENDIF
70    CONTINUE
60    CONTINUE
C
      ENDIF
C
      RETURN
      END
C
C**********************************************************************
C
C    SUBROUTINE LFLOW :
C                        PERFORMS THE LOAD FLOW STUDY BY NEWTON-RAPHSON
C                        METHOD.
C
C**********************************************************************
C
      SUBROUTINE LFLOW(N,NLBUS,NVBUS,ERROR,KVB,MVAB,NG,NC)
       COMPLEX YP(30,30),V(30),VP,YB(30,30),S(30),SL(30,30)
       COMPLEX SLOSS(30,30),SUMM,SLL,GL(30),GR(30),GRR,GLL
```

```
      CHARACTER*10 B(30),LAB(30),TA(10),TB(10),BUSSC(20),LINE(50,2)
      CHARACTER*5 CC
      CHARACTER*13 TYPE
      REAL THETA(30,30),DELTA(30),ERROR,KVB,MVAB,P(30),Q(30)
      REAL JJ(80,80),A(60,60),JC(60,60),TAP(10),QSC(20)
      REAL MAXLIM(20),MINLIM(20),DSETT(20),IMAG,VV
      REAL MXVOLT(10),MNVOLT(10),TMAX(10),TMIN(10),DTAP(10)
      REAL PGMAX(30),PGMIN(30),FLIM(50),PRSETT,PFLOW
      REAL QGMAX(20),QGMIN(20),VB
      INTEGER N,N1,NBUS,NVBUS,NLBUS,COUNT,NTC,NSC,GN,C,NG,NC
      INTEGER GEN(20)
      INTEGER T1(10),T2(10),SC(20),NSCC
C
      COMMON /MAIN0/V,B,S,LAB,P,SLL,GR,GL,NTC,TA,TB,TAP
      COMMON /MAIN1/YB
      COMMON /MAIN2/YP
      COMMON /MAIN3/NSC,BUSSC,QSC,MAXLIM,MINLIM,DSETT
      COMMON /MAIN5/THETA,DELTA
      COMMON /MAIN6/JJ
      COMMON /MAIN7/MXVOLT,MNVOLT,TMAX,TMIN,DTAP,LINE,PGMAX,PGMIN,
     * FLIM,C,QGMAX,QGMIN
      COMMON /MAIN8/JC
      COMMON /MAIN12/NSCC,SC
C
        AV(VP)=ATAN2(AIMAG(VP),REAL(VP))*57.29578
C
         NBUS=N
C
C
      IF(NTC.NE.0)THEN
      DO 1 I=1,NTC
      DO 2 J=1,NBUS
      IF(TA(I).EQ.B(J))THEN
C
      B(NBUS+I)=TA(I)
      T1(I)=J
      GO TO 222
      ENDIF
2     CONTINUE
222    DO 3 K=1,NBUS
      IF(TB(I).EQ.B(K))THEN
      T2(I)=K
      GO TO 333
      ENDIF
3     CONTINUE
C
333    YB(T1(I),T1(I))=YB(T1(I),T1(I))+(1.-TAP(I)**2)*YB(T1(I),T2(I))
C
       YP(T1(I),T2(I))=YP(T1(I),T2(I))-YB(T1(I),T2(I))*(TAP(I)*
     * (TAP(I)-1))
C
       YP(T2(I),T1(I))=YP(T2(I),T1(I))-YB(T2(I),T1(I))*(1-TAP(I))
```

```
C
          YB(T1(I),T2(I))=TAP(I)*YB(T1(I),T2(I))
          YB(T2(I),T1(I))=YB(T1(I),T2(I))
1         CONTINUE
          ENDIF
C
          DO 10 I=1,NBUS
           DO 20 J=1,NBUS
            IF(CABS(YB(I,J)).NE.0.0)THEN
             IF(REAL(YB(I,J)).EQ.0.0.AND.AIMAG(YB(I,J)).GT.0.0)THEN
            THETA(I,J)=3.14159/2.0
             ELSEIF(REAL(YB(I,J)).EQ.0.0.AND.AIMAG(YB(I,J)).LT.0.0)THEN
            THETA(I,J)=-3.14159/2.0
            ELSE
            THETA(I,J)=AV(YB(I,J))/57.29578
            ENDIF
            ELSE
            THETA(I,J)=0.0
            ENDIF
20       CONTINUE
            DELTA(I)=AV(V(I))/57.29578
C
10       CONTINUE
C
         CALL LBUS(COUNT,V,S,NBUS,ERROR,NLBUS,NVBUS,P,
        * GL,B,NTC,TAP,T1,T2)
C
C####################################################################
C  CALCULATING THE LINE POWERS( MEASURED AT SENDING END)
C####################################################################
C
            DO 60 I=1,NBUS
            DO 60 J=1,NBUS
             IF(I.EQ.J.OR.CABS(YB(I,J)).EQ.0.0)GO TO 65
C
             SL(I,J)=V(I)*(CONJG((V(I)-V(J)))*CONJG(-YB(I,J))
        *+CONJG(V(I))*CONJG(YP(I,J)))
C
             GO TO 60
65       SL(I,J)=(0.0,0.0)
60       CONTINUE
C
C
C********  CALCULATE SLACK BUS POWER  ***********
C
             S(1)=SLL
            DO 70 J=2,NBUS
             S(1)=S(1)+SL(1,J)
70         CONTINUE
             GR(1)=S(1)*MVAB
C
C  CALCULATE LINE LOSS
```

```
C
         DO 80 I=1,NBUS
          DO 80 J=1,NBUS
          IF(I.EQ.J)GO TO 90
          SLOSS(I,J)=SL(I,J)+SL(J,I)
          GO TO 80
90       SLOSS(I,J)=(0.0,0.0)
80      CONTINUE
C
C  CALCULATING THE REACTIVE POWER FOR VOLTAGE CONTROL BUSES
C
         IF(NVBUS.NE.0)THEN
         DO 77 I=2,NVBUS+1
         SUM2=0.0
        DO 13 J=1,NBUS
         SUM2=SUM2+CABS(YB(I,J))*CABS(V(J))*CABS(V(I))*SIN(-THETA(I,J)
     * -DELTA(J)+DELTA(I))
13      CONTINUE
         Q(I)=SUM2+AIMAG(GL(I))
         S(I)=CMPLX(P(I),Q(I))
         GL(I)=GL(I)*MVAB
         GR(I)=S(I)*MVAB
77      CONTINUE
          ENDIF
C
         IF(NSC.NE.0) THEN
         NSCC=0
         DO 444 I=1,NSC
         DO 555 J=1,NBUS
          IF(BUSSC(I).EQ.B(J))THEN
C          QSC(I)=QSC(I)*(CABS(V(J)))**2
          IF(LAB(J).EQ.'PQ') THEN
           IMAG=AIMAG(GR(J))+QSC(I)
           GR(J)=CMPLX(REAL(GR(J)),IMAG)
           GR(J)=GR(J)*MVAB
           NSCC=NSCC+1
          SC(NSCC)=J
          ENDIF
          GO TO 444
          ENDIF
555     CONTINUE
444     CONTINUE
          ENDIF
C
C
C*********************************************************************
C    OUTPUT 'INITIAL.DAT' FOR EXPERT SYSTEM (VOLTAGE & VAR CONTROL)
C  'THIS OUTPUT IS THE INITIAL LOAD FLOW RESULTS'
C*********************************************************************
C
       OPEN (UNIT=7,FILE='INITIAL.DAT')
C      WRITE(7,*)KVB,MVAB
```

```
      DO 950 I=1,NBUS
      VB=CABS(V(I))*KVB
C     GRR=GR(I)/MVAB
C     GLL=GL(I)/MVAB
950   WRITE(7,960)B(I),LAB(I),VB,AV(V(I)),GR(I),GL(I)
      WRITE(7,*)'EOF-INITIAL'
960   FORMAT(A,1X,A3,1X,6(F8.4,1X))
      CLOSE(7)
C
C*********************************************************************
C   OUTPUT 'ANGLE.DAT' FOR EXPERT SYSTEM (VOLTAGE ANGLE ADJUSTMENT)
C  'THIS OUTPUT IS THE INITIAL VOLTAGE ANGLES FROM LOAD FLOW RESULTS'
C*********************************************************************
C
      OPEN (UNIT=7,FILE='ANGLE.DAT')
      DO 951 I=1,NBUS
      PRSETT=REAL(GR(I))
      IF(PRSETT.GT.0.0)THEN
      WRITE(7,961)B(I),LAB(I),AV(V(I))
      ENDIF
951   CONTINUE
      WRITE(7,*)'EOF-ANGLE'
961   FORMAT(A,1X,A3,1X,F8.4)
      CLOSE(7)

C
C*********************************************************************
C   OUTPUT 'CTRL.DAT' FOR EXPERT SYSTEM (VOLTAGE & VAR CONTROL)
C   'THIS OUTPUT IS THE DATA ASSOCIATED WITH CONTROL VARIABLES'
C*********************************************************************
C
      OPEN (UNIT=7,FILE='CTRL.DAT')
      CCC=0.0
      DO 850 I=1,1+NVBUS
      VV=CABS(V(I))
      MXVOLT(I)=MXVOLT(I)/KVB
      MNVOLT(I)=MNVOLT(I)/KVB
850   WRITE(7,860)I,B(I),'GV','NO',MXVOLT(I),MNVOLT(I),VV,CCC,
     *QGMAX(I),QGMIN(I)
      IF(NSC.NE.0)THEN
      DO 870 I=1,NSC
          QSC(I)=QSC(I)*MVAB
          MAXLIM(I)=MAXLIM(I)*MVAB
          MINLIM(I)=MINLIM(I)*MVAB
      IF(MAXLIM(I).LT.0.0)THEN
      TYPE='SR'
      ELSE
      TYPE='SC'
      ENDIF
      J=1+NVBUS+I
      WRITE(7,860)J,BUSSC(I),TYPE,'NO',MAXLIM(I),MINLIM(I),QSC(I),
     *DSETT(I),CCC,CCC
```

```
870     CONTINUE
        ENDIF
        IF(NTC.NE.0)THEN
        DO 880 I=1,NTC
        J=1+NVBUS+NSC+I
880     WRITE(7,860)J,TA(I),'TAP',TB(I),TMAX(I),TMIN(I),TAP(I),DTAP(I)
       *,CCC,CCC
        ENDIF
        WRITE(7,*)'EOF-CTRL'
860     FORMAT(I2,1X,A,1X,A3,1X,A,1X,6(F8.4,1X))
        CLOSE(7)
C
C*********************************************************************
C  PRINTING THE LOAD FLOW RESULTS
C*********************************************************************
C
          WRITE(6,325)
325       FORMAT(//,80('#'),/,25X,'COMPLETE LOAD FLOW RESULTS',/,80('#'))
          WRITE(6,350) COUNT
350       FORMAT(//,5X,'PROGRAM CONVERGED IN',I4,'  ITERATIONS'/)
          WRITE(6,400)
400       FORMAT('0',20X,'BUS VOLT.(KV)',7X,'GENERATION',11X,'LOAD'/
       *,20X,14('-'),6X,12('-'),9X,7('-'),/)
          WRITE(6,500)
500       FORMAT(' ',1X,'BUS',7X,'TYPE',4X,'MAGN.',4X,'ANGLE',5X,'MW',7X,
       *'MVAR',5X,' MW',7X,'MVAR'/,1X,73('-'),/)
          DO 200 I=1,NBUS
           V(I)=V(I)*KVB
           VL=CABS(V(I))/KVB
           QGG=AIMAG(GR(I))
         IF((LAB(I).EQ.'SLK'.AND.QGG.GT.QGMAX(I))
       *.OR.(LAB(I).EQ.'SLK'.AND.QGG.LT.QGMIN(I)))THEN
             WRITE(6,601)B(I),LAB(I),CABS(V(I)),AV(V(I)),GR(I),GL(I)
 601     FORMAT(' ',1X,A10,1X,A3,2(F8.3,1X),2X,F8.3,1X,F8.3,'*',
       *2(F8.3,1X)//)
 C
         ELSEIF((LAB(I).EQ.'PV'.AND.QGG.GT.QGMAX(I))
       *.OR.(LAB(I).EQ.'PV'.AND.QGG.LT.QGMIN(I)))THEN
             WRITE(6,602)B(I),LAB(I),CABS(V(I)),AV(V(I)),GR(I),GL(I)
 602        FORMAT(' ',1X,A10,1X,A3,2(F8.3,1X),2X,F8.3,1X,F8.3,'*',
       *2(F8.3,1X)//)
          ELSEIF((LAB(I).EQ.'PQ'.AND.VL.GT.1.0500001)
       *.OR.(LAB(I).EQ.'PQ'.AND.VL.LT.0.9500001))THEN
            WRITE(6,603)B(I),LAB(I),CABS(V(I)),AV(V(I)),GR(I),GL(I)
 603       FORMAT(' ',1X,A10,1X,A3,F8.3,'*',F8.3,3X,4(F8.3,1X)//)
 C
         ELSE
            WRITE(6,600)B(I),LAB(I),CABS(V(I)),AV(V(I)),GR(I),GL(I)
 600     FORMAT(' ',1X,A10,1X,A3,2(F8.3,1X),2X,4(F8.3,1X)//)
         ENDIF
 200     CONTINUE
 C
```

```
C****** PRINT LINE FLOW **************
C
        WRITE(6,700)
700    FORMAT(80('#'),//,' ',29X,'LINE FLOW',10X,'LINE LOSSES',
      *4X,'FLOW LIM.',/,29X,
      *11('-'),8X,13('-'),2X,11('-'),/)
          WRITE(6,800)
800    FORMAT(' ',6X, 'LINE',17X,' MW ',5X,'MVAR',6X,' MW ',5X,'MVAR'
      *,6X,'MW',/,2X,77('-'),/)
          DO 300 I=1,NBUS
          DO 300 J=1,NBUS
          IF(I.EQ.J.OR.SL(J,I).EQ.(0.0,0.0))GO TO 300
C           SL(J,I)=SL(J,I)*MVAB
            SL(I,J)=SL(I,J)*MVAB
            SLOSS(I,J)=SLOSS(I,J)*MVAB
C*******
       DO 2222 K=1,C
       IF(B(I).EQ.LINE(K,1).AND.B(J).EQ.LINE(K,2)) GO TO 3333
       IF(B(I).EQ.LINE(K,2).AND.B(J).EQ.LINE(K,1)) GO TO 3333
 2222  CONTINUE
C*******
3333   LF=ABS(REAL(SL(I,J)))
        IF(LF.GT.FLIM(K))THEN
          WRITE(6,900)B(I),B(J),SL(I,J),SLOSS(I,J),FLIM(K)
 900      FORMAT(' ',2X,A8,'-TO- ',A8,2X,F8.3,'*',
      *3(F8.3,1X),1X,F8.3)
        ELSE
          WRITE(6,901)B(I),B(J),SL(I,J),SLOSS(I,J),FLIM(K)
 901      FORMAT(' ',2X,A8,'-TO- ',A8,2X,4(F8.3,1X),1X,F8.3)
        ENDIF
 300      CONTINUE
          WRITE(6,1000)
 1000     FORMAT(//,1X,80('#'),//)
C*******************
C
          WRITE(6,551)
 551    FORMAT(//,80('#'),/,20X,'GENERATORS/SYNCHRONOUS CONDENSERS'
      *   ,/,80('#'),//)
          WRITE(6,561)
 561    FORMAT(2X,'LOCATION',2X,'BUS TYPE',4X,'PRES. GEN.',11X,
      *   'GENERATION CAPACITY',/,3X,'(BUS)',15X,'MW',5X,'MVAR',
      *   6X,'MW',7X,'MW',6X,'MVAR',6X,'MVAR',
      *   /,38X,'(MAX.)',3X,'(MIN.)',3X,'(MAX.)',4X,'(MIN.)'
      *   ,/,2X,75('-'),//)
 C
        DO 523 I=1,NBUS
        PRSETT=REAL(GR(I))
        IF(I.EQ.1)THEN
        WRITE(6,535)B(I),LAB(I),REAL(GR(I)),AIMAG(GR(I)),QGMAX(I)
      *,QGMIN(I)
        ELSEIF(I.GT.1.AND.I.LE.NVBUS+1)THEN
        WRITE(6,534)B(I),LAB(I),REAL(GR(I)),AIMAG(GR(I)),PGMAX(I),
```

```
     *PGMIN(I),QGMAX(I),QGMIN(I)
      ENDIF
      IF(I.GT.NVBUS+1.AND.PRSETT.GT.0.0)THEN
       WRITE(6,536)B(I),LAB(I),REAL(GR(I)),AIMAG(GR(I)),PGMAX(I),
     *PGMIN(I)
      ENDIF
534   FORMAT(2X,A8,4X,A3,2X,6(F8.3,1X))
535   FORMAT(2X,A8,4X,A3,2X,2(F8.3,1X),18X,2(F8.3,1X))
536   FORMAT(2X,A8,4X,A3,2X,4(F8.3,1X))
523   CONTINUE
C
C****************************************
         IF(NSC.NE.0) THEN
          WRITE(6,550)
550   FORMAT(//,80('#'),/,25X,'STATIC VAR COMPANSATORS',/,
     *  80('#'),//)
          WRITE(6,560)
560   FORMAT(5X,'LOCATION',10X,'TYPE',8X,'PR. SETT.',6X,
     *  'MVAR LIMITS',/,6X,'(BUS)',25X,'(MVAR)',5X,'(MAX.)',
     *  5X,'(MIN.)',/,3X,62('-'),//)
         DO 570 I=1,NSC
        IF(MAXLIM(I).LT.0.0)THEN
          TYPE='SH. REACTOR'
        ELSE
          TYPE='SH. CAPACITOR'
        ENDIF
         WRITE(6,580)BUSSC(I),TYPE,QSC(I),MAXLIM(I),MINLIM(I)
580   FORMAT(5X,A,3X,A,3X,F8.3,3X,2(F8.3,3X))
570   CONTINUE
         ENDIF
C
         IF(NTC.NE.0)THEN
C
         WRITE(6,100)
100   FORMAT(//,80('#'),/,25X,'TRANSFORMER TAP CHANGERS',/
     *  ,80('#'),//)
         WRITE(6,101)
101   FORMAT(17X,'BETWEEN',13X,'TAP',7X,'TAP',9X,'LIMITS',
     *  /,2X,'TAP ID.',5X,'BUS    -    BUS',8X,'LOCATIN',3X,
     *  'POSITION',2X,'(MAX.)',3X,'(MIN.)',/,1X,73('-'),/)
C
         DO 1111 I=1,NTC
C
          WRITE(6,110)I,TA(I),TB(I),TA(I),TAP(I),TMAX(I),TMIN(I)
110   FORMAT(2X,I2,10X,A8,2X,A8,3X,A8,3X,F6.4,3X,F6.4,3X,F6.4)
1111  CONTINUE
          ENDIF
C
C*******************************************************************
C     OUTPUT 'GEN.DAT' FOR EXPERT SYSTEM (GENERATION RE-SCHEDULING &
C     VOLTAGE ANGLE ADJUSTMENT)
C*******************************************************************
```

123-6

```
C
      OPEN(UNIT=7,FILE='GEN.DAT')
C
      GN=0
      DO 123 I=1,NBUS-1
      PRSETT=REAL(GR(I+1))
      IF(I.LE.NG)THEN
      GN=I
      GEN(I)=I+1
      WRITE(7,234)GN,B(I+1),PGMAX(I+1),PGMIN(I+1),PRSETT
      ENDIF
      IF(I.GT.NVBUS+1.AND.PRSETT.GT.0.0)THEN
      GN=GN+1
      GEN(GN)=I+1
      WRITE(7,234)GN,B(I+1),PGMAX(I+1),PGMIN(I+1),PRSETT
      ENDIF
234   FORMAT(I2,1X,A,1X,3(F8.4,1X))
123   CONTINUE
C
      WRITE(7,*)'EOF-GEN'
C
      CLOSE(7)
C

C
C*********************************************************************
C  OUTPUT 'PFLOW.DAT' FOR:
C  EXPERT SYSTEM (GENERAGTION RE-SCHEDULING)
C*********************************************************************
C
      OPEN(UNIT=7,FILE='PFLOW.DAT')
C
      DO 345 K=1,C
       DO 456 I=1,NBUS
       IF(LINE(K,1).EQ.B(I)) GO TO 567
456    CONTINUE
567    DO 678 J=1,NBUS
       IF(LINE(K,2).EQ.B(J)) GO TO 340
678    CONTINUE
340   PFLOW=REAL(SL(I,J))
       WRITE(7,341)K,LINE(K,1),LINE(K,2),PFLOW,FLIM(K)
341   FORMAT(I2,1X,A,1X,A,1X,F8.4,1X,F8.4)
345   CONTINUE
C
      WRITE(7,*)'EOF-PFLOW'
C
      CLOSE(7)
C
C
C ************************************************************************
C  CALL SUBROUTINE 'SENSIT'(CALCULATES SENSITVITY MATRIX FOR POWER SYS.)
C ************************************************************************
```

```
C
        CALL SENSIT(NBUS,NLBUS,NVBUS,NTC,B,KVB,MVAB,TA)
C
C ********************************************************************
C  CALL SUBROUTINE 'LSENS'( CALCULATES LINE LOADING SENS. MATRIX )
C ********************************************************************
C
      CALL LSENS(NBUS,NVBUS,NLBUS,KVB,MVAB,V,B,LAB,
     *C,LINE,GN,GEN)
C
C*********************************************************************
C
C
          RETURN
          END
C
C*********************************************************************
C  SUBROUTINE LBUS:
C                     THIS SUB. IS USED BY THE SUB. (LFLOW) TO PERFORM
C                     NECESSARY CALCULATIONS.
C
C*********************************************************************
C
      SUBROUTINE LBUS(COUNT,V,S,NBUS,ERROR,NLBUS,NVBUS
     *,P,GL,B,NTC,TAP,T1,T2)
        CHARACTER*10 B(30)
        COMPLEX YB(30,30),V(30),S(30),GL(30)
        REAL THETA(30,30),DELTA(30),SUM1,SUM2,VV(30),P(30),TAP(10)
        REAL DP(30),DQ(30),ERROR,RL,IM,JJ(80,80),JC(60,60)
        INTEGER NBUS,II,COUNT,NLBUS,NVBUS,TT,NTC,T1(10),T2(10)
C
       COMMON /MAIN1/YB
       COMMON /MAIN5/THETA,DELTA
       COMMON /MAIN6/JJ
       COMMON /MAIN8/JC
C
        TT=0
        COUNT=0
       DO 22 I=1,NBUS
22     VV(I)=CABS(V(I))
5      DO 2 I=2,NBUS
        SUM1=0.0
        SUM2=0.0
       DO 3 J=1,NBUS
        SUM1=SUM1+CABS(YB(I,J))*VV(J)*VV(I)*COS(-THETA(I,J)
     * -DELTA(J)+DELTA(I))
        SUM2=SUM2+CABS(YB(I,J))*VV(J)*VV(I)*SIN(-THETA(I,J)
     * -DELTA(J)+DELTA(I))
3       CONTINUE
C
        IF(NVBUS.NE.0.AND.I.GT.1.AND.I.LE.NVBUS+1)THEN
        DP(I)=P(I)-SUM1-REAL(GL(I))
```

```
        DQ(I)=0.0
        ELSE
        DP(I)=REAL(S(I))-SUM1
        DQ(I)=AIMAG(S(I))-SUM2
        ENDIF
C
2       CONTINUE
C
      DO 4 II=2,NBUS
       IF(ABS(DP(II)).LE.ERROR.AND.ABS(DQ(II)).LE.ERROR)THEN
        IF(II.EQ.NBUS)THEN
        DO 44 J=2,NBUS
         RL=VV(J)*COS(DELTA(J))
         IM=VV(J)*SIN(DELTA(J))
        V(J)=CMPLX(RL,IM)
44    CONTINUE
          GO TO 11
        ENDIF
        GO TO 4
       ELSE
C
      CALL JACOB1(NBUS,VV,DP,DQ,NLBUS,NVBUS,TT
     *,B,NTC,TAP,T1,T2)
C
       COUNT=COUNT+1
       GO TO 5
       ENDIF
4     CONTINUE
C
11    TT=1
C
      CALL JACOB1(NBUS,VV,DP,DQ,NLBUS,NVBUS,TT
     *,B,NTC,TAP,T1,T2)
C
      RETURN
      END



C
C*********************************************************************
C  SUBROUTINE JACOB1:
C                     DETERMINES THE JACOBIAN MATRIX FOR L.FLOW STUDY
C
C*********************************************************************
C
      SUBROUTINE JACOB1(NBUS,VV,DP,DQ,NLBUS,NVBUS,
     *TT,B,NTC,TAP,T1,T2)
        CHARACTER*10 B(30)
        COMPLEX YB(30,30)
        REAL JJ(80,80),DELTA(30),THETA(30,30),VV(30),BB(60)
```

```
      REAL DV(30),DDELTA(30),DP(30),DQ(30),JC(60,60),VR,VI
      REAL AA(30,30),TAP(10),JCC(60,60)
      INTEGER NBUS,L,NLBUS,NVBUS,TT,NTC,T1(10),T2(10),N1
C
C
      COMMON /MAIN1/YB
      COMMON /MAIN5/THETA,DELTA
      COMMON /MAIN6/JJ
      COMMON /MAIN8/JCC
C
C
       L=NBUS+1
      DO 10 I=1,NBUS
       JJ(I,I)=0.0
       JJ(I,L)=0.0
       JJ(L,I)=0.0
       JJ(L,L)=0.0
C
      DO 30 K=1,NBUS
       IF(K.NE.I)THEN
C
      JJ(I,I)=JJ(I,I)-VV(I)*VV(K)*CABS(YB(I,K))
     **SIN(DELTA(I)-DELTA(K)-THETA(I,K))
C
      JJ(I,L)=JJ(I,L)+VV(K)*CABS(YB(I,K))*COS(DELTA(I)-
     *  DELTA(K)-THETA(I,K))
C
C
      JJ(L,I)=JJ(L,I)+VV(I)*VV(K)*CABS(YB(I,K))
     * *COS(DELTA(I)-DELTA(K)-THETA(I,K))
C
      JJ(L,L)=JJ(L,L)+VV(K)*CABS(YB(I,K))*SIN(DELTA(I)-
     * DELTA(K)-THETA(I,K))
C
       ENDIF
 30   CONTINUE
C
      JJ(I,L)=JJ(I,L)+2*VV(I)*CABS(YB(I,I))*COS(THETA(I,I))
C
      JJ(L,L)=JJ(L,L)-2*VV(I)*CABS(YB(I,I))*SIN(THETA(I,I))
C
      DO 20 J=1,NBUS
       IF(J.NE.I)THEN
C
      JJ(I,J)=VV(I)*VV(J)*CABS(YB(I,J))
     * *SIN(DELTA(I)-DELTA(J)-THETA(I,J))
C
      JJ(I,NBUS+J)=VV(I)*CABS(YB(I,J))
     * *COS(DELTA(I)-DELTA(J)-THETA(I,J))
C
      JJ(L,J)=-1*VV(I)*VV(J)*CABS(YB(I,J))
```

```
     * *COS(DELTA(I)-DELTA(J)-THETA(I,J))
C
        JJ(L,NBUS+J)=VV(I)*CABS(YB(I,J))
     * *SIN(DELTA(I)-DELTA(J)-THETA(I,J))
C
          ENDIF
20     CONTINUE
C
          L=L+1
10     CONTINUE
C
        DO 44 I=1,NBUS-1
44      BB(I)=DP(I+1)
        DO 444 I=1,NLBUS
         BB(NBUS-1+I)=DQ(NVBUS+1+I)
444    CONTINUE
        DO 41 I=1,NBUS-1
         DO 42 J=1,NBUS-1
        JC(I,J)=JJ(I+1,J+1)
        JCC(I,J)=JJ(I+1,J+1)
42     CONTINUE
41     CONTINUE
        DO 43 I=1,NBUS-1-NVBUS
         DO 45 J=1,NBUS-1
        JC(NBUS-1+I,J)=JJ(NBUS+1+NVBUS+I,J+1)
        JCC(NBUS-1+I,J)=JJ(NBUS+1+NVBUS+I,J+1)
45      CONTINUE
43      CONTINUE
        DO 51 I=1,NBUS-1
         DO 52 J=1,NLBUS
        JC(I,NBUS-1+J)=JJ(I+1,NBUS+1+NVBUS+J)
        JCC(I,NBUS-1+J)=JJ(I+1,NBUS+1+NVBUS+J)
52      CONTINUE
51      CONTINUE
        DO 53 I=1,NLBUS
         DO 54 J=1,NLBUS
        JC(NBUS-1+I,NBUS-1+J)=JJ(NBUS+1+NVBUS+I,NBUS+1+NVBUS+J)
        JCC(NBUS-1+I,NBUS-1+J)=JJ(NBUS+1+NVBUS+I,NBUS+1+NVBUS+J)
54      CONTINUE
53      CONTINUE
C
          NN=2*(NBUS-1)-NVBUS
C
         IF (TT.EQ.1)THEN
C
       IF(NTC.NE.0)THEN
       N1=2*NBUS
       DO 60 I=1,N1
       DO 65 J=1,NTC
       JJ(I,N1+J)=0.0
65     CONTINUE
60     CONTINUE
```

123-6

```
      DO 70 I=1,NTC
      G1=ABS(REAL(YB(T1(I),T2(I))))/TAP(I)
      B1=ABS(AIMAG(YB(T1(I),T2(I))))/TAP(I)
C
      JJ(T1(I),N1+I)=2*VV(T1(I))**2*G1*TAP(I)+VV(T1(I))*VV(T2(I))*
     * (-G1*COS(DELTA(T1(I))-DELTA(T2(I)))+B1*SIN(DELTA(T1(I))-
     * DELTA(T2(I))))
C
      JJ(T2(I),N1+I)=VV(T1(I))*VV(T2(I))*
     * (-G1*COS(DELTA(T2(I))-DELTA(T1(I)))+B1*SIN(DELTA(T2(I))-
     * DELTA(T1(I))))
C
      JJ(NBUS+T1(I),N1+I)=2*VV(T1(I))**2*B1*TAP(I)+VV(T1(I))*VV(T2(I)
     * )*(-B1*COS(DELTA(T1(I))-DELTA(T2(I)))-G1*SIN(DELTA(T1(I))-
     * DELTA(T2(I))))
C
      JJ(NBUS+T2(I),N1+I)=VV(T1(I))*VV(T2(I))*
     * (-B1*COS(DELTA(T2(I))-DELTA(T1(I)))-G1*SIN(DELTA(T2(I))-
     * DELTA(T1(I))))
C
70    CONTINUE
       ENDIF
       GO TO 1001
       ENDIF
C
          CALL FACTOR(JC,NN,BB)
C
           DO 55 I=2,NBUS
55     DDELTA(I)=BB(I-1)
           DO 56 I=1,NLBUS
            DV(I+NVBUS+1)=BB(NBUS-1+I)
56    CONTINUE
C
           DO 15 I=2,NBUS
15     DELTA(I)=DELTA(I)+DDELTA(I)
C
            DO 16 I=NVBUS+2,NBUS
            VV(I)=VV(I)+DV(I)
16    CONTINUE
1001  RETURN
           END
C
C***********************************************************************
C  SUBROUTINE SENSIT:
C                     DETERMINES THE SENSITIVITY TABLE OF POWER SYSTEM
C
C***********************************************************************
C
       SUBROUTINE SENSIT(NBUS,NLBUS,NVBUS,NTC,BBB,KVB,MVAB,TA)
       CHARACTER*10 BBB(30),TA(10)
       CHARACTER*15 COL(80),ROW(80)
       INTEGER NBUS,NLBUS,NVBUS,N,N1,DIM,L,M,NN,MM,NTC,N2,NSCC,SC(20)
```

```
      INTEGER SC1(20)
      REAL SENS(80,80),A(1,60),B(1,30),J1(60,60)
      REAL D(60,30),E(30,60),F(30,30),JC(60,60)
      REAL A1(1,60),B1(1,30)
      REAL D1(60,30),E1(30,60),F1(30,30)
      REAL BB(1,30),FF(30,30),EE
      REAL JJ(80,80),ANGSEN,MVAB,KVB
C
       COMMON /MAIN6/JJ
       COMMON /MAIN8/JC
       COMMON /MAIN12/NSCC,SC
C
      NN=2*(NBUS-1)-NVBUS
      N=2*NBUS
      N1=NVBUS+1
      N2=N1+NTC
C
C
      L=NLBUS
      M=NBUS-1
      MM=M+L
      EE=0.0001
C
      DO 5 I=1,NBUS-1
5     A(1,I)=JJ(1,I+1)
C
      DO 6 I=1,NLBUS
6     A(1,NBUS-1+I)=JJ(1,NBUS+NVBUS+1+I)
C
      DO 7 I=1,NVBUS+1
7     B(1,I)=JJ(1,NBUS+I)
C
      IF(NTC.NE.0)THEN
      DO 8 I=1,NTC
8     B(1,N1+I)=JJ(1,N+I)
      ENDIF
C
      DO 10 I=1,NBUS-1
      DO 20 J=1,NVBUS+1
      D(I,J)=JJ(I+1,NBUS+J)
20    CONTINUE
10    CONTINUE
C
       IF(NTC.NE.0)THEN
       DO 11 I=1,NBUS-1
       DO 21 J=1,NTC
       D(I,N1+J)=JJ(I+1,N+J)
21    CONTINUE
11    CONTINUE
       ENDIF
C
       DO 30 I=1,NLBUS
```

```
      DO 40 J=1,NVBUS+1
      D(NBUS-1+I,J)=JJ(NBUS+NVBUS+1+I,NBUS+J)
40    CONTINUE
30    CONTINUE
C
      IF(NTC.NE.0)THEN
      DO 31 I=1,NLBUS
      DO 41 J=1,NTC
      D(NBUS-1+I,N1+J)=JJ(NBUS+NVBUS+1+I,N+J)
41    CONTINUE
31    CONTINUE
      ENDIF
C
      DO 50 I=1,NVBUS+1
      DO 60 J=1,NBUS-1
      E(I,J)=JJ(NBUS+I,J+1)
60    CONTINUE
50    CONTINUE
C
      DO 70 I=1,NVBUS+1
      DO 80 J=1,NLBUS
      E(I,NBUS-1+J)=JJ(NBUS+I,NBUS+NVBUS+1+J)
80    CONTINUE
70    CONTINUE
C
      DO 90 I=1,NVBUS+1
      DO 95 J=1,NVBUS+1
      F(I,J)=JJ(NBUS+I,NBUS+J)
95    CONTINUE
90    CONTINUE
C
      IF(NTC.NE.0)THEN
      DO 91 I=1,NVBUS+1
      DO 96 J=1,NTC
      F(I,N1+J)=JJ(NBUS+I,N+J)
96    CONTINUE
91    CONTINUE
      ENDIF
C
      CALL INVERT(JC,J1,NN)
C
      CALL MULMAT(A,J1,A1,1,NN,NN,NN,1,NN,1,60,60,60,1,60)
C
      CALL MULMAT(A1,D,BB,1,NN,NN,N2,1,N2,1,60,60,30,1,30)
C
      CALL MULMAT(J1,D,D1,NN,NN,NN,N2,NN,N2,60,60,60,30,60,30)
C
      CALL MULMAT(E,J1,E1,N1,NN,NN,NN,N1,NN,30,60,60,60,30,60)
C
      CALL MULMAT(E,D1,FF,N1,NN,NN,N2,N1,N2,30,60,60,30,30,30)
C
      CALL SUBMAT(B,BB,B1,1,N2,1,30,1,30,1,30)
```

123-6

```
C
      CALL SUBMAT(F,FF,F1,N1,N2,30,30,30,30,30,30)
C
      DO 51 K=1,NN
51     SENS(1,K)=A1(1,K)
      DO 101 K=1,NVBUS+1+NTC
      IF(K.LE.N1) THEN
        SENS(1,NN+K)=B1(1,K)*MVAB
        ENDIF
C
        IF(K.GT.N1) THEN
        SENS(1,NN+K)=B1(1,K)*MVAB
        ENDIF
101   CONTINUE
C
      DO 67 J=1,NN
      DO 68 K=1,NN
       IF(J.LE.M) THEN
        SENS(J+1,K)=J1(J,K)*57.29578/MVAB
       ENDIF
C
       IF(J.GT.M) THEN
        SENS(J+1,K)=J1(J,K)/MVAB
       ENDIF
C
68     CONTINUE
67     CONTINUE
C
        DO 100 J=1,NN
         DO 110 K=1,NVBUS+1+NTC
        IF(J.LE.M.AND.K.LE.N1)THEN
         SENS(J+1,NN+K)=-1*D1(J,K)*57.29578
         ENDIF
C
         IF(J.LE.M.AND.K.GT.N1)THEN
         SENS(J+1,NN+K)=-1*D1(J,K)*57.29578
         ENDIF
C
         IF(J.GT.M.AND.K.LE.N1)THEN
         SENS(J+1,NN+K)=-1*D1(J,K)
         ENDIF
C
         IF(J.GT.M.AND.K.GT.N1)THEN
         SENS(J+1,NN+K)=-1*D1(J,K)
         ENDIF
110       CONTINUE
100      CONTINUE
C
         DO 120 J=1,NVBUS+1
         DO 130 K=1,NN
          SENS(NN+1+J,K)=E1(J,K)
130     CONTINUE
```

123-6

```
 120    CONTINUE
C
        DO 140 J=1,NVBUS+1
        DO 150 K=1,NVBUS+1+NTC
        IF(K.LE.N1)THEN
        SENS(NN+1+J,NN+K)=F1(J,K)*MVAB
        ENDIF
C
        IF(K.GT.N1)THEN
        SENS(NN+1+J,NN+K)=F1(J,K)*MVAB
        ENDIF
150     CONTINUE
140     CONTINUE
C
        COL(1)='DP(  '//BBB(1)
        DO 201 J=2,NBUS
201      COL(J)='DD(  '//BBB(J)
C
        DO 202 J=1,NLBUS
202      COL(NBUS+J)='DVL(  '//BBB(N1+J)
C
        DO 203 J=1,N1
203      COL(NBUS+NLBUS+J)='DQG(  '//BBB(J)
C
        DO 210 J=1,NBUS-1
210      ROW(J)='DP(  '//BBB(J+1)
C
        DO 220 J=1,NLBUS
220      ROW(NBUS-1+J)='DQL(  '//BBB(N1+J)
C
        DO 230 J=1,N1
230      ROW(NBUS-1+NLBUS+J)='DVG(  '//BBB(J)
C
        IF(NTC.NE.0)THEN
        DO 240 J=1,NTC
240      ROW(NBUS-1+NBUS+J)='DT(  '//TA(J)
        ENDIF
C
C  SENS = COMPLETE SENSITIVITY MATRIX
C
C
C*********************************************************************
C    OUTPUT 'SENSIT.DAT' FOR EXPERT SYSTEM (VOLTAGE & VAR CONTROL)
C*********************************************************************
C
        OPEN (UNIT=7,FILE='SENSIT.DAT')
C
        COL(1)='DP-SLK '//BBB(1)
         DO 301 J=2,NBUS
301     COL(J)='D-DELTA '//BBB(J)
C
         DO 302 J=1,NLBUS
```

123-6

```
302    COL(NBUS+J)='DV-LOAD '//BBB(N1+J)
C
        DO 303 J=1,N1
303    COL(NBUS+NLBUS+J)='DQ-GEN '//BBB(J)
C
        DO 310 J=1,NBUS-1
310    ROW(J)='DP '//BBB(J+1)
C
        L=0
       DO 320 J=1,NLBUS
       ROW(NBUS-1+J)='DQ-LOAD '//BBB(N1+J)
        IF(L.LT.NSCC)THEN
        DO 315 K=1,NSCC
        IF(BBB(N1+J).EQ.BBB(SC(K)))THEN
        L=L+1
        SC1(L)=J
        GO TO 320
        ENDIF
315    CONTINUE
        ENDIF
320    CONTINUE
C
        DO 330 J=1,N1
330    ROW(NBUS-1+NLBUS+J)='DV-GEN '//BBB(J)
C
        IF(NTC.NE.0)THEN
        DO 340 J=1,NTC
340    ROW(NBUS-1+NBUS+J)='D-TAP '//TA(J)
        ENDIF
C
       DO 402 K=1,NSCC
       DO 402 J=1,NLBUS+NVBUS+1
       WRITE(7,502)COL(J+NBUS),ROW(SC1(K)+NBUS-1)
      *,SENS(J+NBUS,SC1(K)+NBUS-1)
402    CONTINUE
502    FORMAT(2(A,1X),F10.6)
C
       DO 400 K=1,N+NTC-NBUS-NLBUS
       DO 400 J=1,NLBUS+NVBUS+1
       WRITE(7,500)COL(J+NBUS),ROW(K+NBUS-1+NLBUS)
      *,SENS(J+NBUS,K+NBUS-1+NLBUS)

C
400    CONTINUE
       WRITE(7,*)'EOF-SENSIT'
500    FORMAT(2(A,1X),F10.4)
       CLOSE(7)
C
C
C*********************************************************************
C   OUTPUT 'ANGSEN.DAT' FOR EXPERT SYSTEM (VOLTAGE ANGLE ADJUSTMENT)
C*********************************************************************
```

123-6

```
C
C
      OPEN (UNIT=7,FILE='ANGSEN.DAT')
C
      DO 401 K=1,NVBUS
      DO 401 J=1,NVBUS
      ANGSEN=SENS(J+1,K)
      WRITE(7,501)COL(J+1),ROW(K),ANGSEN
401   CONTINUE
      WRITE(7,*)'EOF-ANGSEN'
501   FORMAT(2(A,1X),F9.4)
      CLOSE(7)
C
      RETURN
      END
C
C
C ***********************************************************************
C    SUBROUTINE " LSENS "
C                        DETERMINE THE LINE FLOW SENSITIVITY MATRIX
C ***********************************************************************
C
C
      SUBROUTINE LSENS(NBUS,NVBUS,NLBUS,KVB,MVAB,V,B,LAB,
     *NLINE,LINE,GN,GEN)
C
      COMPLEX V(30),YB(30,30),YP(30,30)
      CHARACTER*10 B(30),LINE(50,2),LAB(30),B1(40,2)
      REAL DELTA(30),THETA(30,30),HH(60),SUM1,SUM2,VV(30)
      REAL KVB,MVAB,RL,IM
      REAL HH1(40,30),H(30)
      REAL JC(60,60)
      INTEGER NBUS,NVBUS,NLBUS,NLINE,NN,KK,GN,GEN(20)
C
        COMMON /MAIN1/YB
        COMMON /MAIN2/YP
        COMMON /MAIN5/THETA,DELTA
        COMMON /MAIN8/JC
        COMMON /LSENS1/HH,H
C
      NN=NBUS-1+NLBUS
      KK=NBUS-1
C
      DO 55 L=1,NBUS
      V(L)=V(L)/KVB
      VV(L)=CABS(V(L))
55    CONTINUE
C
C
      DO 5 K=1,NLINE
      DO 10 I=1,NBUS
      IF(LINE(K,1).EQ.B(I)) GO TO 20
```

```
 10     CONTINUE
 20     DO 30 J=1,NBUS
        IF(LINE(K,2).EQ.B(J)) GO TO 40
 30     CONTINUE
C
C ***********************************************************************
C    CALCULATIONS
C ***********************************************************************
C
 40     DO 50 L=1,NN
        HH(L)=0.0
 50     CONTINUE
C
C
        IF(LAB(I).EQ.'SLK'.AND.LAB(J).EQ.'PV')THEN
        HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
       *DELTA(J)-THETA(I,J))
        GO TO 1121
        ENDIF
C
        IF(LAB(I).EQ.'PV'.AND.LAB(J).EQ.'SLK')THEN
        HH(I-1)=-ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
       *DELTA(J)-THETA(I,J))
        GO TO 1121
        ENDIF
C
        IF(LAB(I).EQ.'SLK'.AND.LAB(J).EQ.'PQ')THEN
        HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
       *DELTA(J)-THETA(J,I))
C
C
        HH(NBUS-2-NVBUS+J)=ABS(VV(I)*CABS(YB(I,J)))*COS(DELTA(I)-
       *DELTA(J)-THETA(I,J))
C
        GO TO 1121
        ENDIF
C
        IF(LAB(I).EQ.'PQ'.AND.LAB(J).EQ.'SLK')THEN
        HH(I-1)=-ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
       *DELTA(J)-THETA(I,J))
C
        HH(NBUS-2-NVBUS+I)=ABS(VV(J)*CABS(YB(I,J)))*COS(DELTA(I)-
       *DELTA(J)-THETA(I,J))-2*ABS(VV(I)*CABS(YB(I,J)))
       *COS(THETA(I,J))
C
        GO TO 1121
        ENDIF
C
        IF(LAB(I).EQ.'PV'.AND.LAB(J).EQ.'PQ')THEN
        HH(I-1)=-ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
       *DELTA(J)-THETA(I,J))
```

123-6

```
C
       HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(NBUS-2-NVBUS+J)=ABS(VV(I)*CABS(YB(I,J)))*COS(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       GO TO 1121
       ENDIF
C
       IF(LAB(I).EQ.'PQ'.AND.LAB(J).EQ.'PV')THEN
C
       HH(I-1)=-ABS(VV(J)*VV(I)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(NBUS-2-NVBUS+I)=ABS(VV(J)*CABS(YB(I,J)))*COS(DELTA(I)-
     *DELTA(J)-THETA(I,J))-2*ABS(VV(I)*CABS(YB(I,J)))*COS(THETA(J,I))
C
       GO TO 1121
       ENDIF
C
       IF(LAB(I).EQ.'PQ'.AND.LAB(J).EQ.'PQ')THEN
C
       HH(I-1)=-ABS(VV(J)*VV(I)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(NBUS-2-NVBUS+I)=ABS(VV(J)*CABS(YB(I,J)))*COS(DELTA(I)-
     *DELTA(J)-THETA(I,J))-2*ABS(VV(I)*CABS(YB(I,J)))*COS(THETA(J,I))
C
       HH(NBUS-2-NVBUS+J)=ABS(VV(I)*CABS(YB(I,J)))*COS(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       GO TO 1121
       ENDIF
C
       IF(LAB(I).EQ.'PV'.AND.LAB(J).EQ.'PV')THEN
C
       HH(I-1)=-ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       HH(J-1)=ABS(VV(I)*VV(J)*CABS(YB(I,J)))*SIN(DELTA(I)-
     *DELTA(J)-THETA(I,J))
C
       GO TO 1121
       ENDIF
```

```
C
C
1121  CALL LSENSX(NN,I,J,B,NBUS,NVBUS,NLBUS)
C
       B1(K,1)=B(I)
       B1(K,2)=B(J)
       DO 11 L=1,KK
11     HH1(K,L)=H(L)
C
5      CONTINUE
C
C
C*********************************************************************
C  OUTPUT 'LSENSIT.DAT' FOR EXPERT SYSTEM (GENERATION RE-SCHEDULING)
C*********************************************************************
C
       OPEN(UNIT=7,FILE='LSENSIT.DAT')
C
       DO 222 I=1,GN
       DO 333 J=1,NLINE
       K=GEN(I)-1
       WRITE(7,444)J,B(GEN(I)),HH1(J,K)
333    CONTINUE
222    CONTINUE
       WRITE(7,*)'EOF-LSENSIT'
444    FORMAT(I2,1X,A,1X,F8.4)
       CLOSE(7)

C
       RETURN
       END
C
C *********************************************************************
C  SUBROUTINE  LSENSX
C *********************************************************************
C
       SUBROUTINE LSENSX(NN,I,J,B,NBUS,NVBUS,NLBUS)
C
       CHARACTER*10 LAB(30),B(30)
       REAL HH(60),HH1(60,30),H(30),BB(4)
       REAL E(60),JC(60,60),BB2(4),JJ(60,60)
       INTEGER NBUS,NVBUS,NLBUS,NN,KK
C
        COMMON /LSENS1/HH,H
        COMMON /MAIN8/JC
C
C
c      DO 222 L=1,NBUS-1
       DO 222 L=1,NVBUS
       DO 800 M=1,NN
800    E(M)=0.0
       E(L)=1.0
```

123-6

```
C
      DO 960 M=1,NN
      DO 970 N=1,NN
      JJ(M,N)=JC(M,N)
970   CONTINUE
960   CONTINUE
      CALL FACTOR(JJ,NN,E)
      DO 980 M=1,NN
980   HH1(M,L)=E(M)
C
222   CONTINUE
      KK=NBUS-1
C
      CALL MULMAT1(HH,HH1,H,NN,NN,KK,KK)
C
      RETURN
      END
C
C*********************************************************************
C     SUBROUTINE FACTOR:
C
C
C                SUBROUTINE FOR TRIANGULER FACTERIZATHION AND SOLUTION
C
C*********************************************************************
C
      SUBROUTINE FACTOR(JJ,NN,BB)
      REAL JJ(60,60),BB(60)
      INTEGER NN
      CALL LUG(JJ,NN)
C
      CALL SOLVE(JJ,NN,BB)
C
       RETURN
      END
C
C*********************************************************************
C
      SUBROUTINE LUG(JJ,NN)
      REAL JJ(60,60)
       IF (NN.EQ.1)RETURN
        NM1=NN-1
         DO 10 K=1,NM1
          KP1=K+1
           DO 10 J=KP1,NN
            T=JJ(K,J)/JJ(K,K)
           JJ(K,J)=T
          DO 10 I=KP1,NN
10       JJ(I,J)= JJ(I,J)-T*JJ(I,K)
         RETURN
        END
```

```
C########################################################################
C
        SUBROUTINE SOLVE(JJ,NN,BB)
C
C* ROUTINE SOLVES  A.X=B WITH RESULT X RETURNED IN B WHICH IS
C* DESTROYED ON IN INPUT A CONTAINS LU FACTORS OBTAINED FROM LUG,
C* LUROW OR CROUT. IT HAS DIMENSION N WITH THE ABSOLUTE ROW DIMENSION
C* OF A BEING IA IN THE CALLING PROGRAMME.
C
        REAL JJ(60,60),BB(NN)
C
C* FORWARD SUBSTITUTION
C
        BB(1)=BB(1)/JJ(1,1)
         IF (NN.EQ.1)RETURN
          DO 20 I=2,NN
           T=BB(I)
            IM1=I-1
             DO 10 J=1,IM1
10           T=T-JJ(I,J)*BB(J)
20            BB(I)=T/JJ(I,I)
C
C* BACK SUBSTITUTION
C
        NM1=NN-1
         DO 40 II=1,NM1
          I=NN-II
           IP1=I+1
            T=BB(I)
             DO 30 J=IP1,NN
30         T=T-JJ(I,J)*BB(J)
40        BB(I)=T
          RETURN
         END
C
C  ************************************************************************
C     SUBROUTINE SUBMAT(C=A-B)
C  ************************************************************************
C
       SUBROUTINE SUBMAT(A,B,C,NR,NC,A1,A2,B1,B2,C1,C2)
       INTEGER NR,NC,A1,A2,B1,B2,C1,C2
       REAL A(A1,A2),B(B1,B2),C(C1,C2)
C
       DO 10 I=1,NR
       DO 20 J=1,NC
       C(I,J)=A(I,J)-B(I,J)
20    CONTINUE
10    CONTINUE
C
       RETURN
       END
C
```

```
C  ********************************************************************
C     SUBROUTINE MULMAT(C=A*B)
C  ********************************************************************
C
       SUBROUTINE MULMAT(A,B,C,NRA,NCA,NRB,NCB,NRC,NCC,A1,A2,B1,B2
     *,C1,C2)
       INTEGER NRA,NCA,NRB,NCB,NRC,NCC,A1,A2,B1,B2,C1,C2
       REAL A(A1,A2),B(B1,B2),C(C1,C2)
C
       DO 10 I=1,NRC
       DO 20 J=1,NCC
       C(I,J)=0.0
       DO 30 K=1,NRB
       C(I,J)=C(I,J)+A(I,K)*B(K,J)
30    CONTINUE
20    CONTINUE
10    CONTINUE
       RETURN
       END

C
C  ********************************************************************
C     SUBROUTINE MULMAT1(C=A*B)
C  ********************************************************************
C
       SUBROUTINE MULMAT1(A,B,C,NCA,NRB,NCB,NCC)
       INTEGER NCA,NRB,NCB,NCC
       REAL A(60),B(60,30),C(30)
C
       DO 10 I=1,NCC
       C(I)=0.0
      DO 30 K=1,NRB
      C(I)=C(I)+A(K)*B(K,I)
30    CONTINUE
10    CONTINUE
       RETURN
       END
C
C  ********************************************************************
C     SUBROUTINE INVERT(B=INV(A))
C  ********************************************************************
C
       SUBROUTINE INVERT(A,B,N)
       REAL A(60,60),B(60,60)
        DO 10 I=1,N
        DO 15 J=1,N
        B(I,J)=A(I,J)
15    CONTINUE
10    CONTINUE
C
       DO 1 IP=1,N
       DO 2 IR=1,N
```

```
      IF(IR.EQ.IP)GO TO 2
      DO 3 IC=1,N
      IF(IC.EQ.IP)GO TO 3
      B(IR,IC)=B(IR,IC)-(B(IR,IP)*B(IP,IC)/B(IP,IP))
3     CONTINUE
2     CONTINUE
      B(IP,IP)=-1./B(IP,IP)
      DO 4 I=1,N
      IF(I.EQ.IP)GO TO 4
      B(I,IP)=B(I,IP)*B(IP,IP)
      B(IP,I)=B(IP,I)*B(IP,IP)
4     CONTINUE
1     CONTINUE
      DO 5 IR=1,N
      DO 5 IC=1,N
5     B(IR,IC)=-B(IR,IC)
      RETURN
      END
C
C***********************************************************************
C           END OF THE PROGRAM
C***********************************************************************
C
```

123-6

APPENDIX - B

COMPUTER LISTING OF

THE PROTOTYPE E.S.

APPENDIX - B.1

COMPUTER LISTING FOR KB OF

VOLTAGE & VAR ADJUSTMENT PROCESS

KNOWLEDGE BASE FOR VOLTAGE AND VAR CONTROL

```
(defglobal ;/ GLOBAL VARIABLES
     ?*count* = 1
     ?*countt* = 0
     ?*countt1* = 0
     ?*max-sen* = 0.0
     ?*max-cont* = fyh
     ?*max-dev* = 0.0
     ?*bus-dev* = fyh
     ?*max-type* = type
     ?*max-dev* = 0.0
     ?*g-count* = 0
     ?*old-bus* = fyh
     ?*old-busq* = fyh
     ?*max-dsr1* = 0.0
     ?*max-dsr2* = 0.0)


(deftemplate sensit          ;/ DATA BASE FOR SENSITIVITY COEFFICIENTS
     (slot dep-var (default ?NONE))
     (slot bus-name (default ?NONE))
     (slot ctrl-var (default ?NONE))
     (slot ctrl-loc (default ?NONE))
     (slot sen-fact (default ?NONE)))

(deftemplate control         ;/ DATA BASE FOR CONTROL ELEMENTS
     (slot ctrl-num (default ?NONE))
     (slot location (default ?NONE))
     (slot ctrl-typ (default ?NONE))
     (slot sec-term (default ?NONE))
     (slot max-lim (default ?NONE))
     (slot min-lim (default ?NONE))
     (slot pr-set (default ?NONE))
     (slot ch-pr-st (default ?NONE))
     (slot qgmax (default ?NONE))
     (slot qgmin (default ?NONE))
     (slot prior (default ?NONE)))


(deftemplate in-state        ;/ DATA BASE FOR INITIAL L. FLOW RESULTS
     (slot bus-name (type SYMBOL) (default ?NONE))
     (slot bus-type (type SYMBOL) (default ?NONE))
     (slot bus-volt (default ?NONE))
     (slot q-gen (default ?NONE))
     (slot pr-dev (type NUMBER) (default 0))
     (slot priorty (type NUMBER) (default 0))
     (slot seq (type NUMBER) (default 0)))
```

```
(defrule open-file-initial      ;This rule opens the file which contains
    (initial-fact)              ;the initial L. Flow results
=>
    (open "initial.dat" initial "r")
    (open "ctrl.dat" ctrl "r")
    (assert (read-file-initial)
            (last-round-in)
            (last-last-r)
            (read-file-ctrl)
            (nlbus 0)
            (nbus 0)
            (ngen 0)))


(defrule read-file-initial                    ;This rule reads the initial L. Flow
    ?nbuss <- (nbus ?nbus)                    ;results and stores in the Expert
    ?nlbuss <- (nlbus ?nlbus)                 ;System (ES) Data-Base
    ?read-initial <- (read-file-initial)
=>
    (retract ?read-initial)
    (bind ?bus-name (read initial))
    (bind ?bus-type (read initial))
    (bind ?bus-volt (read initial))
    (bind ?volt-ang (read initial))
    (bind ?p-gen (read initial))
    (bind ?q-gen (read initial))
    (bind ?p-load (read initial))
    (bind ?q-load (read initial))
    (if (neq ?bus-name EOF-INITIAL)
    then
    (bind ?pr-dev 0.0)
    (retract ?nbuss)
    (assert (nbus =(+ ?nbus 1)))
    (if (eq ?bus-type PQ)
    then
    (if (> ?bus-volt 1.05)
    then
    (bind ?pr-dev (- ?bus-volt 1.05)))
    (if (< ?bus-volt 0.95)
    then
    (bind ?pr-dev (- ?bus-volt 0.95)))
    (if (and (<= ?bus-volt 1.05) (>= ?bus-volt 0.95))
    then
    (bind ?pr-dev 0.0))
    (retract ?nlbuss)
    (assert (nlbus =(+ ?nlbus 1))))
    (assert (in-state (bus-name ?bus-name) (bus-type ?bus-type)
                      (bus-volt ?bus-volt)
                      (q-gen ?q-gen) (pr-dev ?pr-dev)))
    (assert (read-file-initial))
    ) ;close 'if'
```

123-6

```
    (if (eq ?bus-name EOF-INITIAL)
    then
    (assert (EOF-INITIAL))
    (close initial))) ;close the file


(defrule read-file-ctrlit                              ;This rule reads the file
    ?read-ctrl <- (read-file-ctrl)                     ;which contains the data of
    ?ngenn <- (ngen ?ngen)
=>                                                     ;Control Equipment and stores
    (retract ?read-ctrl)                               ;it in the ES Data-Base
    (bind ?ctrl-num (read ctrl))
    (bind ?location (read ctrl))
    (bind ?ctrl-typ (read ctrl))
    (bind ?sec-term (read ctrl))
    (bind ?max-lim (read ctrl))
    (bind ?min-lim (read ctrl))
    (bind ?pr-set (read ctrl))
    (bind ?ch-pr-st (read ctrl))
    (bind ?qgmax (read ctrl))
    (bind ?qgmin (read ctrl))
    (if (neq ?ctrl-num EOF-CTRL)
    then
    (if (eq ?ctrl-typ GV)
    then
    (retract ?ngenn)
    (assert (ngen =(+ ?ngen 1))))
    (assert (control (ctrl-num ?ctrl-num) (location ?location)
                     (ctrl-typ ?ctrl-typ) (sec-term ?sec-term)
                     (max-lim ?max-lim)
                     (min-lim ?min-lim) (pr-set ?pr-set)
                     (ch-pr-st ?ch-pr-st)
                     (qgmax ?qgmax) (qgmin ?qgmin) (prior 1)))
    (assert (read-file-ctrl))
    ) ;close 'if'
    (if (eq ?ctrl-num EOF-CTRL)
    then
    (assert (EOF-CTRL))
    (assert (q-dev-check))
    (close ctrl)))


(defrule qgen-violation  ;/ THIS RULE CHECKS VIOLATION IN Q-GENERATION
    ?ngenn <- (ngen ?ngen)
    ?qq-dev <- (q-dev-check)
    ?q-gener <- (in-state (bus-name ?bus-name)
                          (bus-type SLK|PV)
                          (q-gen ?q-gen)
                          (pr-dev ?pr-dev) (seq 0))
    ?ctrl <- (control (location ?bus-name)
                      (ctrl-typ GV)
                      (qgmax ?qgmax) (qgmin ?qgmin))
```

```
    (test (or (< ?q-gen ?qgmin)
              (> ?q-gen ?qgmax)))
=>
    (retract ?qq-dev)
    (bind ?*g-count* (+ ?*g-count* 1))
    (if (< ?q-gen ?qgmin)
    then
    (bind ?pr-dev (- ?q-gen ?qgmin)))
    (if (> ?q-gen ?qgmax)
    then
    (bind ?pr-dev (- ?q-gen ?qgmax)))
    (if (and (>= ?q-gen ?qgmin) (<= ?q-gen ?qgmax))
    then
    (bind ?pr-dev 0.0))
    (modify ?q-gener (pr-dev ?pr-dev) (seq 1))
    (if (> ?ngen ?*g-count*)
    then
    (assert (q-dev-check))
    ) ; close 'if'
    ) ; close the rule


(defrule volt-violation-check                       ;This rule scans all the volt-
     ?initial <- (initial-fact)                     ;ages of load buses and lets
     ?voltage <- (in-state (bus-name ?bus-name)     ;those rules associated with
           (bus-type PQ) (bus-volt ?bus-volt)       ;voltage adjustment to fire, if
           (pr-dev ?pr-dev&~0.0))                   ;there is any violation in
                                                    ;load bus voltages
=>
     (retract ?initial)
     (assert (violation-exist)))


(defrule qgen-violation-check                       ;This rule scans all the q-gen.
     ?initial <- (initial-fact)                     ;of gen. buses and lets
                                                    ;those rules associated with
     ?q-gener <- (in-state (bus-name ?bus-name)     ;q-gen. adjustment to fire, if
              (bus-type SLK|PV)                     ;there is any violation in
     (q-gen ?q-gen)                                 ;q-generation.
(pr-dev ?pr-dev&~0.0))
=>
     (retract ?initial)
     (assert (violation-exist)))


(defrule print-no-violation                         ;This rule prints NO-VIOLATION
     (declare (salience -5))                        ;message if all the bus volt-
     (not (violation-exist))                        ;ages within the acceptable
=>    ;limits
     (printout t " THERE IS NO VIOLATIONS IN LOAD BUS VOLTAGES AND
```

123-6

```
      Q-GENERATION " crlf))

(defrule open-file-violation                  ;This rule opens the two files
    (violation-exist)                         ;which contains the Sensitivity
=>                                            ;Factors and the Control Equip-
;                                             ;ment Data
    (open "sensit.dat" sensit "r")
    (assert (read-file-sensit)))

(defrule read-file-sensit                     ;This rule reads the file
    ?read-sens <- (read-file-sensit)          ;which contains the Sensitivity
=>                                            ;Factors and stores them in
    (retract ?read-sens)                      ;the ES Data-Base
    (bind ?dep-var (read sensit))
    (bind ?bus-name (read sensit))
    (bind ?ctrl-var (read sensit))
    (bind ?ctrl-loc (read sensit))
    (bind ?sen-fact (read sensit))
    (if (neq ?dep-var EOF-SENSIT)
    then
    (assert (sensit (dep-var ?dep-var) (bus-name ?bus-name)
                    (ctrl-var ?ctrl-var)
                    (ctrl-loc ?ctrl-loc) (sen-fact ?sen-fact)))
    (assert (read-file-sensit))
    ) ;close 'if'
    (if (eq ?dep-var EOF-SENSIT)
    then
    (assert (EOF-SENSIT))
    (close sensit)))

(defrule under-voltage-check-1  ;/ This rule & the next one identify the
    (EOF-INITIAL)               ;/ most under voltage bus
    (EOF-SENSIT)
    (EOF-CTRL)
    ?voltage <- (in-state (bus-name ?bus-name)
                          (bus-type PQ)
                          (bus-volt ?bus-volt&:(< ?bus-volt 0.95))
                          (priorty ?priort&0))
    (not (update-1))
    (not (last-round))
=>
    (if (eq ?*old-bus* ?bus-name) ; 'if-1'
    then
    (assert (max-under-volt))
    (assert (update-1))
    (bind ?*bus-dev* ?bus-name)
    else
    (assert (max-under-volt))
    (bind ?pr-dev (- ?bus-volt 0.95))
```

```
(bind ?*max-dev* (min ?*max-dev* ?pr-dev))
(if (= ?*max-dev* ?pr-dev)
then
(bind ?*bus-dev* ?bus-name)
) ; close 'if'
) ; close 'if-1'
) ; close the rule


(defrule under-voltage-check-2
     (declare (salience -1))
     ?maxx <- (max-under-volt)
=>
     (retract ?maxx)
     (bind ?bus-name ?*bus-dev*)
     (bind ?*max-dev* 0.0)
     (bind ?*bus-dev* fyh)
     (assert (under-volt-1 ?bus-name)
             (control-bus ?bus-name type 0)
             (maxx 0.0)
             (update-1)))


(defrule over-voltage-check-1  ;/ This rule & the next one identify the
                               ;/ most over voltage bus
     (declare (salience -50))
     (EOF-INITIAL)
     (EOF-SENSIT)
     (EOF-CTRL)
     ?voltage <- (in-state (bus-name ?bus-name)
                           (bus-type PQ)
                           (bus-volt ?bus-volt&:(> ?bus-volt 1.05))
                           (priorty ?priorty&0))
     (not (update-1))
     (not (last-round))
=>
     (if (eq ?*old-bus* ?bus-name) ; 'if-1'
     then
     (assert (max-over-volt))
     (assert (update-1))
     (bind ?*bus-dev* ?bus-name)
     else
     (assert (max-over-volt))
     (bind ?pr-dev (- ?bus-volt 1.05))
     (bind ?*max-dev* (max ?*max-dev* ?pr-dev))
     (if (= ?*max-dev* ?pr-dev)
     then
     (bind ?*bus-dev* ?bus-name)
     ) ; close 'if'
     ) ; close 'if-1'
```

```
) ; close the rule


(defrule over-voltage-check-2
     (declare (salience -51))
     ?maxx <- (max-over-volt)
=>
     (retract ?maxx)
     (bind ?bus-name ?*bus-dev*)
     (bind ?*max-dev* 0.0)
     (bind ?*bus-dev* fyh)
     (assert (over-volt ?bus-name)
             (control-bus ?bus-name type 0)
             (maxx 0.0)
             (update-1)))


(defrule under-qgen-check-1  ;/ This rule & the next one identify the
                             ;/ most under q-gen. bus
     (declare (salience -60))
     (EOF-INITIAL)
     (EOF-SENSIT)
     (EOF-CTRL)
     ?qgen <- (in-state (bus-name ?bus-name)
                        (bus-type SLK|PV)
                        (q-gen ?q-gen) (priorty ?priorty&0))
     ?ctrl <- (control (location ?bus-name)
                       (ctrl-typ GV)
                       (qgmin ?qgmin))
     (not (update-1))
     (not (last-round))
     (test (< ?q-gen ?qgmin))
=>
     (if (eq ?*old-bus* ?bus-name) ; 'if-1'
     then
     (assert (max-under-qgen))
     (assert (update-1))
     (bind ?*bus-dev* ?bus-name)
     else
     (assert (max-under-qgen))
     (bind ?pr-dev (- ?q-gen ?qgmin))
     (bind ?*max-dev* (min ?*max-dev* ?pr-dev))
     (if (= ?*max-dev* ?pr-dev)
     then
     (bind ?*bus-dev* ?bus-name)
     ) ; close 'if'
     ) ; close 'if-1'
     ) ; close the rule
```

123-6

```
(defrule under-qgen-check-2
     (declare (salience -61))
     ?maxx <- (max-under-qgen)
=>
     (retract ?maxx)
     (bind ?bus-name ?*bus-dev*)
     (bind ?*max-dev* 0.0)
     (bind ?*bus-dev* fyh)
     (assert (under-qgen ?bus-name)
             (control-bus ?bus-name type 0)
             (update-1)))


(defrule over-qgen-check-1   ;/ This rule & the next one identify the
                             ;/ most over q-gen. bus
     (declare (salience -62))
     (EOF-INITIAL)
     (EOF-SENSIT)
     (EOF-CTRL)
     ?qgen <- (in-state (bus-name ?bus-name)
                        (bus-type SLK|PV)
                        (q-gen ?q-gen) (priorty ?priorty&0))
     ?ctrl <- (control (location ?bus-name)
                       (ctrl-typ GV)
                       (qgmax ?qgmax))
     (not (update-1))
     (not (last-round))
     (test (> ?q-gen ?qgmax))
=>
     (if (eq ?*old-bus* ?bus-name) ; 'if-1'
     then
     (assert (max-over-qgen))
     (assert (update-1))
     (bind ?*bus-dev* ?bus-name)
     else
     (assert (max-over-qgen))
     (bind ?pr-dev (- ?q-gen ?qgmax))
     (bind ?*max-dev* (max ?*max-dev* ?pr-dev))
     (if (= ?*max-dev* ?pr-dev)
     then
     (bind ?*bus-dev* ?bus-name)
     ) ; close 'if'
     ) ; close 'if-1'
     ) ; close the rule


 (defrule over-qgen-check-2
      (declare (salience -63))
      ?maxx <- (max-over-qgen)
```

```
=>
     (retract ?maxx)
     (bind ?bus-name ?*bus-dev*)
     (bind ?*max-dev* 0.0)
     (bind ?*bus-dev* fyh)
     (assert (over-qgen ?bus-name)
             (control-bus ?bus-name type 0)
             (update-1)))


(defrule last-round          ;/ This rule & next rules reset the buses
    (declare (salience -63));/ for last round of correction proccess
     (last-round-in)
     ?initial <- (in-state (bus-name ?bus-name)
                           (pr-dev ?pr-dev&~0.0)
                           (priorty ?priorty&1))
=>
     (assert (last-round))
     (modify ?initial (priorty 0)))


(defrule confirm-last-round
     (declare (salience -64))
     ?lasti <- (last-round-in)
     ?last <- (last-round)
=>
     (bind ?*old-bus* fyh)
     (retract ?last)
     (retract ?lasti)
     (assert (last-round-ok)))


(defrule new-bus-check       ;/ This rule checks for new violation bus
     (declare (salience 5))
     (or
     ?buss <- (under-volt-1 ?bus-name)
     ?buss <- (over-volt ?bus-name)
     ?buss <- (under-qgen ?bus-name)
     ?buss <- (over-qgen ?bus-name)
     ) ; close 'or'
     ?ctrl <- (control (location ?ctrl-loc)
                       (prior ?prior&~0))
=>
     (if (neq ?bus-name ?*old-bus*)
     then
     (modify ?ctrl (prior 0))
     ) ; close 'if'
     ) ; close the rule
```

```
(defrule change-bus-priority-v  ;/ This rule changes the old load bus if the
    (declare (salience -42))    ;/ violation can not be eliminated
    (or
    ?under-v <- (under-volt-1 ?bus-name)
    ?under-v <- (over-volt-1 ?bus-name)
    ) ; close 'or'
    ?cont <- (control-bus ? ? ?)
    ?max1 <- (maxx 0.0)
    ?update <- (update-1)
    ?initial <- (in-state (bus-name ?bus-name)
                          (priorty ?priorty))
=>
    (retract ?under-v)
    (retract ?cont)
    (retract ?max1)
    (retract ?update)
    (modify ?initial (priorty 1))
    ) ; close the rule


(defrule change-bus-priority-q ;/ This rule changes the old gen. bus if the
                               ;/ violation can not be eliminated
    (declare (salience -62))
    (or
    ?under-v <- (under-qgen ?bus-name)
    ?under-v <- (over-qgen ?bus-name)
    ) ; close 'or'
    ?cont <- (control-bus ? ? ?)
    ?max1 <- (maxx 0.0)
    ?update <- (update-1)
    ?initial <- (in-state (bus-name ?bus-name)
                          (priorty ?priorty))
=>
    (retract ?under-v)
    (retract ?cont)
    (retract ?max1)
    (retract ?update)
    (modify ?initial (priorty 1))
    ) ; close the rule


(defrule under-voltage-local-reactor                  ;This rule adjusts the load
    ?under-volt-1 <- (under-volt-1 ?bus-name)         ;bus voltage if its value
                                                      ;below the limit ,using local
    ?control-bus <- (control-bus ?ctrl ?type&~GV 0);Shunt Reactor(s),if exist
    ?voltage <- (in-state (bus-name ?bus-name)
                (bus-volt ?bus-volt&:(< ?bus-volt 0.95)))
    ?control <- (control (ctrl-num ?ctrl-num)
                         (location ?ctrl)
                         (ctrl-typ ?ctrl-typ&SR)
```

```
                              (max-lim ?max-lim)
                              (min-lim ?min-lim)
                              (pr-set ?pr-set&:(< ?pr-set ?min-lim))
                              (ch-pr-st ?ch-pr-st)
                              (prior ?prior&0))
   ?sensit <- (sensit (dep-var DV-LOAD)
                            (bus-name ?bus-name)
                            (ctrl-var ?ctrl-var&DQ-LOAD)
                            (ctrl-loc ?ctrl)
                            (sen-fact ?sen-fact))
=>
   (bind ?*old-bus* ?bus-name)
   (retract ?under-volt-1)
   (retract ?control-bus)
   (bind ?d-volt (- 1.0 ?bus-volt))
   (bind ?d-sr (/ ?d-volt ?sen-fact))
   (if (<= ?d-sr (- ?min-lim ?pr-set)) ; if-2
   then
   (if (neq ?ch-pr-st 0.0)             ; if-3
   then
   (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
   (bind ?nstep-rq (round ?nstep-rq))
   (bind ?d-sr (* ?nstep-rq ?ch-pr-st))
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   else ; of 'if-3'
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   ) ;close 'if-3'
   else ; of 'if-2'
   (if (neq ?ch-pr-st 0.0)             ; if-5
   then
   (bind ?nstep-av (integer (/(- ?min-lim ?pr-set) ?ch-pr-st)))
   (bind ?d-sr (* ?nstep-av ?ch-pr-st))
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   else ; of 'if-5'
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   ) ;close 'if-5'
   ) ;close 'if-2'
   ) ;close the rule
```

```
(defrule over-voltage-local-reactor                    ;This rule adjusts the load
    ?under-volt-1 <- (over-volt ?bus-name)             ;bus voltage if its value
                                                       ;above the limit ,using local
    ?control-bus <- (control-bus ?ctrl ?type&~GV 0);Shunt Reactor(s),if exist
    ?voltage <- (in-state (bus-name ?bus-name)
                  (bus-volt ?bus-volt&:(> ?bus-volt 1.05)))
    ?control <- (control (ctrl-num ?ctrl-num)
                          (location ?ctrl)
                          (ctrl-typ ?ctrl-typ&SR)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set&:(> ?pr-set ?max-lim))
                          (ch-pr-st ?ch-pr-st)
                          (prior ?prior&0))
    ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var&DQ-LOAD)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-fact))
=>
     (bind ?*old-bus* ?bus-name)
     (retract ?under-volt-1)
     (retract ?control-bus)
     (bind ?d-volt (- 1.05 ?bus-volt))
     (bind ?d-sr (/ ?d-volt ?sen-fact))
     (if (>= ?d-sr (- ?max-lim ?pr-set)) ; if-2
     then
     (if (neq ?ch-pr-st 0.0)               ; if-3
     then
     (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
     (bind ?nstep-rq (round ?nstep-rq))
     (bind ?d-sr (* ?nstep-rq ?ch-pr-st))
     (assert (d-chang ?d-sr))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     else ; of 'if-3'
     (assert (d-chang ?d-sr))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     ) ;close 'if-3'
     else ; of 'if-2'
     (if (neq ?ch-pr-st 0.0)               ; if-5
     then
     (bind ?nstep-av (integer (/(- ?max-lim ?pr-set) ?ch-pr-st)))
     (bind ?d-sr (* ?nstep-av ?ch-pr-st))
     (assert (d-chang ?d-sr))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
```

```
else ; of 'if-5'
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-5'
) ;close 'if-2'
) ;close the rule

(defrule under-voltage-local-capacitor          ;This rule adjusts the load
   (declare (salience -5))                      ;bus voltage if its value
                                                ;below the limit ,using local
   ?under-volt-1 <- (under-volt-1 ?bus-name)   ;Shunt Capacitor(s),if exist
   ?control-bus <- (control-bus ?ctrl ?type&~GV 0|1)
   ?voltage <- (in-state (bus-name ?bus-name)
                  (bus-volt ?bus-volt&:(< ?bus-volt 0.95)))
   ?control <- (control (ctrl-num ?ctrl-num)
                        (location ?ctrl)
                        (ctrl-typ ?ctrl-typ&SC)
                        (max-lim ?max-lim)
                        (min-lim ?min-lim)
                        (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                        (ch-pr-st ?ch-pr-st)
                        (prior ?prior&0))
   ?sensit <- (sensit (dep-var DV-LOAD)
                      (bus-name ?bus-name)
                      (ctrl-var ?ctrl-var&DQ-LOAD)
                      (ctrl-loc ?ctrl)
                      (sen-fact ?sen-fact))
=>
   (bind ?*old-bus* ?bus-name)
   (retract ?under-volt-1)
   (retract ?control-bus)
   (bind ?d-volt (- 1.0 ?bus-volt))
   (bind ?d-sr (/ ?d-volt ?sen-fact))
   (if (<= ?d-sr (- ?max-lim ?pr-set)) ; if-2
   then
   (if (neq ?ch-pr-st 0.0)             ; if-3
   then
   (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
   (bind ?nstep-rq (round ?nstep-rq))
   (bind ?d-sr (* ?nstep-rq ?ch-pr-st))
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   else ; of 'if-3'
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
```

123-6

```
    ) ;close 'if-3'
    else ; of 'if-2'
    (if (neq ?ch-pr-st 0.0)              ; if-5
    then
    (bind ?nstep-av (integer (/(- ?max-lim ?pr-set) ?ch-pr-st)))
    (bind ?d-sr (* ?nstep-av ?ch-pr-st))
    (assert (d-chang ?d-sr))
    (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
    (assert (next-stage-1a))
    (assert (next-stage-1b))
    else ; of 'if-5'
    (assert (d-chang ?d-sr))
    (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
    (assert (next-stage-1a))
    (assert (next-stage-1b))
    ) ;close 'if-5'
    ) ;close 'if-2'
    ) ;close the rule


(defrule over-voltage-local-capacitor          ;This rule adjusts the load
    (declare (salience -5))                    ;bus voltage if its value
                                               ;above the limit ,using local
    ?under-volt-1 <- (over-volt ?bus-name)     ;Shunt Capacitor(s),if exist
    ?control-bus <- (control-bus ?ctrl ?type&~GV 0|1)
    ?voltage <- (in-state (bus-name ?bus-name)
                 (bus-volt ?bus-volt&:(> ?bus-volt 1.05)))
    ?control <- (control (ctrl-num ?ctrl-num)
                         (location ?ctrl)
                         (ctrl-typ ?ctrl-typ&SC)
                         (max-lim ?max-lim)
                         (min-lim ?min-lim)
                         (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                         (ch-pr-st ?ch-pr-st)
                         (prior ?prior&0))
    ?sensit <- (sensit (dep-var DV-LOAD)
                       (bus-name ?bus-name)
                       (ctrl-var ?ctrl-var&DQ-LOAD)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-fact))
=>
    (bind ?*old-bus* ?bus-name)
    (retract ?under-volt-1)
    (retract ?control-bus)
    (bind ?d-volt (- 1.05 ?bus-volt))
    (bind ?d-sr (/ ?d-volt ?sen-fact))
    (if (>= ?d-sr (- ?min-lim ?pr-set)) ; if-2
    then
    (if (neq ?ch-pr-st 0.0)             ; if-3
    then
    (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
    (bind ?nstep-rq (round ?nstep-rq))
```

123-6

```
(bind ?d-sr (* ?nstep-rq ?ch-pr-st))
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
else ; of 'if-3'
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-3'
else ; of 'if-2'
(if (neq ?ch-pr-st 0.0)              ; if-5
then
(bind ?nstep-av (integer (/(- ?min-lim ?pr-set) ?ch-pr-st)))
(bind ?d-sr (* ?nstep-av ?ch-pr-st))
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
else ; of 'if-5'
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-5'
) ;close 'if-2'
) ;close the rule


(defrule under-voltage-local-tap-changer        ;This rule adjusts the load
   (declare (salience -10))                     ;bus voltage if its value
                                                ;below the limit ,using local
    ?under-volt-1 <- (under-volt-1 ?bus-name)   ;Tap Changer(s),if exist
    ?control-bus <- (control-bus ?ctrl ?type&~GV ?)
    ?voltage <- (in-state (bus-name ?bus-name)
                (bus-volt ?bus-volt&:(< ?bus-volt 0.95)))
    ?control <- (control (ctrl-num ?ctrl-num)
                         (location ?ctrl)
                         (ctrl-typ ?ctrl-typ&TAP)
                         (sec-term ?sec-term)
                         (max-lim ?max-lim)
                         (min-lim ?min-lim)
                         (pr-set ?pr-set)
                         (ch-pr-st ?ch-pr-st)
                         (prior ?prior&0))
    ?voltage-1 <- (in-state (bus-name ?sec-term)
                            (bus-type ?type-1&PQ)
                            (bus-volt ?volt-1))
    ?voltage-2 <- (in-state (bus-name ?ctrl)
                            (bus-volt ?volt-2))
    ?sensit <- (sensit (dep-var DV-LOAD)
```

```
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var&D-TAP)
                        (ctrl=loc ?ctrl)
                        (sen-fact ?sen-fact))
    ?sensit-1 <- (sensit (dep-var DV-LOAD)
                        (bus-name ?sec-term)
                        (ctrl-var D-TAP)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-1))
    ?sensit-2 <- (sensit (dep-var DV-LOAD)
                        (bus-name ?ctrl)
                        (ctrl-var D-TAP)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-2))
(test (or (and (eq ?ctrl ?bus-name) (> ?pr-set ?min-lim) (> ?volt-1 0.96))
          (and (neq ?ctrl ?bus-name) (eq ?bus-name ?sec-term)
               (< ?pr-set ?max-lim))
          (and (neq ?ctrl ?bus-name) (neq ?bus-name ?sec-term)
               (> ?pr-set ?min-lim))))
=>
     (bind ?*old-bus* ?bus-name)
     (retract ?under-volt-1)
     (retract ?control-bus)
     (bind ?d-volt (- 1.0 ?bus-volt))
     (bind ?d-sr (/ ?d-volt ?sen-fact))
     (bind ?d-sr-1 (   (- 0.96 ?volt-1) ?sen-1))
     (bind ?d-sr-2 (   (- 0.96 ?volt-2) ?sen-2))
     (if (or (and (< ?d-sr 0.0) (>= ?d-sr (- ?min-lim ?pr-set))
                  (>= ?d-sr ?d-sr-1))
             (and (> ?d-sr 0.0) (<= ?d-sr (- ?max-lim ?pr-set))
                  (<= ?d-sr ?d-sr-2)))                           ; if-2
     then
     (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
     (bind ?nstep-rq (integer ?nstep-rq))
     (if (= ?nstep-rq 0) ; 'if-4a'
     then
     (modify ?control (prior 1))
     (assert (under-volt-1 ?bus-name)
             (control-bus ?bus-name type 2))
     else
     (bind ?d-sr (* ?nstep-rq ?ch-pr-st))
     (assert (d-chang ?d-sr))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     ) ;close 'if-4a'
     else ; of 'if-2'
     (if (< ?d-sr 0.0)                        ; if-5
     then
     (if (>= (- ?min-lim ?pr-set) ?d-sr-1)    ; if-6
     then
     (bind ?nstep-av (integer (/(- ?min-lim ?pr-set) ?ch-pr-st)))
```

```
(bind ?pr-set ?min-lim)
else
(bind ?nstep-av (integer (/ ?d-sr-1 ?ch-pr-st)))
(bind ?pr-set (+ ?pr-set (* ?nstep-av ?ch-pr-st)))
) ; close 'if-6'
(if (= ?nstep-av 0) ; 'if-6a'
then
(modify ?control (prior 1))
(assert (under-volt-1 ?bus-name)
        (control-bus ?bus-name type 2))
else
(bind ?d-sr (* ?nstep-av ?ch-pr-st))
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ; close 'if-6a'
else
(if (<= (- ?max-lim ?pr-set) ?d-sr-2)      ; if-7
then
(bind ?nstep-av (integer (/(- ?max-lim ?pr-set) ?ch-pr-st)))
(bind ?pr-set ?max-lim)
else
(bind ?nstep-av (integer (/ ?d-sr-2 ?ch-pr-st)))
(bind ?pr-set (+ ?pr-set (* ?nstep-av ?ch-pr-st)))
) ; close 'if-7'
(if (= ?nstep-av 0) ; 'if-7a'
then
(modify ?control (prior 1))
(assert (under-volt-1 ?bus-name)
        (control-bus ?bus-name type 2))
else
(bind ?d-sr (* ?nstep-av ?ch-pr-st))
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-7a'
)  ;close  'if-5'
) ;close 'if-2'
) ;close the rule


(defrule over-voltage-local-tap-changer            ;This rule adjusts the load
   (declare (salience -10))                        ;bus voltage if its value
   ?under-volt-1 <- (over-volt ?bus-name)          ;above the limit ,using local
   ?control-bus <- (control-bus ?ctrl ?type&~GV ?) ;Tap Changer(s),if exist
   ?voltage <- (in-state (bus-name ?bus-name)

   (bus-volt ?bus-volt&:(> ?bus-volt 1.05)))
   ?control <- (control (ctrl-num ?ctrl-num)
                        (location ?ctrl)
```

```
                          (ctrl-typ ?ctrl-typ&TAP)
                          (sec-term ?sec-term)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (ch-pr-st ?ch-pr-st)
                          (prior ?prior&0))
    ?voltage-1 <- (in-state (bus-name ?sec-term)
                            (bus-type ?type-1&PQ)
                            (bus-volt ?volt-1))
    ?voltage-2 <- (in-state (bus-name ?ctrl)
                            (bus-volt ?volt-2))
    ?sensit <- (sensit (dep-var DV-LOAD)
                       (bus-name ?bus-name)
                       (ctrl-var ?ctrl-var&D-TAP)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-fact))
    ?sensit-1 <- (sensit (dep-var DV-LOAD)
                       (bus-name ?sec-term)
                       (ctrl-var D-TAP)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-1))
    ?sensit-2 <- (sensit (dep-var DV-LOAD)
                       (bus-name ?ctrl)
                       (ctrl-var D-TAP)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-2))
(test (or (and (eq ?ctrl ?bus-name) (< ?pr-set ?max-lim) (< ?volt-1 1.05))
          (and (neq ?ctrl ?bus-name) (eq ?bus-name ?sec-term)
               (> ?pr-set ?min-lim))
          (and (neq ?ctrl ?bus-name) (neq ?bus-name ?sec-term)
               (< ?pr-set ?max-lim))))
=>
     (bind ?*old-bus* ?bus-name)
     (retract ?under-volt-1)
     (retract ?control-bus)
     (bind ?d-volt (- 1.05 ?bus-volt))
     (bind ?d-sr (/ ?d-volt ?sen-fact))
     (bind ?d-sr-1 (/ (- 1.05 ?volt-1) ?sen-1))
     (bind ?d-sr-2 (/ (- 1.05 ?volt-2) ?sen-2))
     (if (or (and (> ?d-sr 0.0) (<= ?d-sr (- ?max-lim ?pr-set))
                  (<= ?d-sr ?d-sr-1))
             (and (< ?d-sr 0.0) (>= ?d-sr (- ?min-lim ?pr-set))
                  (>= ?d-sr ?d-sr-2)))                            ; if-2
     then
     (bind ?nstep-rq (/ ?d-sr ?ch-pr-st))
     (bind ?nstep-rq (integer ?nstep-rq))
     (if (= ?nstep-rq 0) ; 'if-4a'
     then
     (bind ?nstep-rq 1)
     (bind ?d-sr (* ?nstep-rq ?ch-pr-st))
     (assert (d-chang ?d-sr))
```

```
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
else
(bind ?d-sr (* ?nstep-rq ?ch-pr-st))
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-4a'
else ; of 'if-2'
(if (> ?d-sr 0.0)                          ; if-5
then
(if (<= (- ?max-lim ?pr-set) ?d-sr-1)      ; if-6
then
(bind ?nstep-av (integer (/(- ?max-lim ?pr-set) ?ch-pr-st)))
(bind ?pr-set ?max-lim)
else
(bind ?nstep-av (integer (/ ?d-sr-1 ?ch-pr-st)))
(bind ?pr-set (+ ?pr-set (* ?nstep-av ?ch-pr-st)))
) ; close 'if-6'
(if (= ?nstep-av 0) ; 'if-6a'
 then
 (modify ?control (prior 1))
 (assert (under-volt-1 ?bus-name)
         (control-bus ?bus-name type 2))
 else
 (bind ?d-sr (* ?nstep-av ?ch-pr-st))
 (assert (d-chang ?d-sr))
 (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
 (assert (next-stage-1a))
 (assert (next-stage-1b))
 ) ; close 'if-6a'
 else
 (if (>= (- ?min-lim ?pr-set) ?d-sr-2)     ; if-7
 then
 (bind ?nstep-av (integer (/(- ?min-lim ?pr-set) ?ch-pr-st)))
 (bind ?pr-set ?min-lim)
 else
 (bind ?nstep-av (integer (/ ?d-sr-2 ?ch-pr-st)))
 (bind ?pr-set (+ ?pr-set (* ?nstep-av ?ch-pr-st)))
 ) ; close 'if-7'
(if (= ?nstep-av 0) ; 'if-7a'
  then
  (modify ?control (prior 1))
  (assert (under-volt-1 ?bus-name)
          (control-bus ?bus-name type 2))
  else
  (bind ?d-sr (* ?nstep-av ?ch-pr-st))
  (assert (d-chang ?d-sr))
  (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
  (assert (next-stage-1a))
```

```
    (assert (next-stage-1b))
    ) ; close 'if-7a'
    )  ;close  'if-5'
    ) ;close 'if-2'
    ) ;close the rule


(defrule uv-nearby-var-source  ;/ This rule determines the most effective
     (declare (salience -20))  ;/ usable nearby var source for under voltage
                               ;/ correction
     ?under-volt <- (under-volt-1 ?bus-name)
     ?control <- (control (location ?loc&~?bus-name)
                          (ctrl-typ ?type&SR|SC)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (prior ?prior&0))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var DQ-LOAD)
                        (ctrl-loc ?loc)
                        (sen-fact ?sen-fact))
(test (or (and (eq ?type SR) (< ?pr-set ?min-lim))
          (and (eq ?type SC) (< ?pr-set ?max-lim))))
=>
     (assert (max-var))
     (bind ?*max-sen* (max ?sen-fact ?*max-sen*))
     (if (= ?*max-sen* ?sen-fact)
     then
      (bind ?*max-cont* ?loc)
      (bind ?*max-type* ?type)))



(defrule ov-nearby-var-source  ;/ This rule determines the most effective
                               ;/ usable nearby var source for over voltage
                               ;/ correction
     (declare (salience -20))
     ?under-volt <- (over-volt ?bus-name)
     ?control <- (control (location ?loc&~?bus-name)
                          (ctrl-typ ?type&SR|SC)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (prior ?prior&0))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var DQ-LOAD)
                        (ctrl-loc ?loc)
                        (sen-fact ?sen-fact))
 (test (or (and (eq ?type SR) (> ?pr-set ?max-lim))
           (and (eq ?type SC) (> ?pr-set ?min-lim))))
```

```
=>
     (assert (max-var))
     (bind ?*max-sen* (max ?sen-fact ?*max-sen*))
     (if (= ?*max-sen* ?sen-fact)
     then
      (bind ?*max-cont* ?loc)
      (bind ?*max-type* ?type)))


(defrule new-nearby-max-cont-var
     (declare (salience -25))
     ?max1 <- (max-var)
     ?control-bus <- (control-bus ? ? ?)
=>
     (retract ?max1)
     (retract ?control-bus)
     (assert (control-bus ?*max-cont* ?*max-type* 0))
     (bind ?*max-sen* 0.0)
     (bind ?*max-cont* fyh)
     (bind ?*max-type* type))


(defrule uv-nearby-tap-changer  ;/ This rule determines the most effective
                                ;/ usable nearby tap changer for under
                                ;/ voltage correction
     (declare (salience -30))
     ?under-volt <- (under-volt-1 ?bus-name)
     ?control <- (control (location ?loc&~?bus-name)
                          (ctrl-typ ?type&TAP)
                          (sec-term ?sec-term)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (prior ?prior&0))
     ?voltage-1 <- (in-state (bus-name ?sec-term)
                             (bus-type ?type-1&PQ)
                             (bus-volt ?volt-1))
     ?voltage-2 <- (in-state (bus-name ?loc)
                             (bus-volt ?volt-2))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var D-TAP)
                        (ctrl-loc ?loc)
                        (sen-fact ?sen-fact))
 (test (or (and (< ?sen-fact 0.0) (> ?pr-set ?min-lim) (> ?volt-1 0.96))
           (and (> ?sen-fact 0.0) (< ?pr-set ?max-lim) (> ?volt-2 0.96))))
 =>
     (assert (max-tap))
     (if (< ?sen-fact 0.0)
     then
     (bind ?sen-fact (* ?sen-fact -1)))
```

```
(bind ?*max-sen* (max ?sen-fact ?*max-sen*))
(if (= ?*max-sen* ?sen-fact)
then
 (bind ?*max-cont* ?loc)
 (bind ?*max-type* ?type)))


(defrule ov-nearby-tap-changer  ;/ This rule determines the most effective
                                ;/ usable nearby tap changer for over
                                ;/ voltage correction
     (declare (salience -30))
     ?under-volt <- (over-volt ?bus-name)
     ?control <- (control (location ?loc&~?bus-name)
                          (ctrl-typ ?type&TAP)
                          (sec-term ?sec-term)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (prior ?prior&0))
     ?voltage-1 <- (in-state (bus-name ?sec-term)
                             (bus-type ?type-1&PQ)
                             (bus-volt ?volt-1))
     ?voltage-2 <- (in-state (bus-name ?loc)
                             (bus-volt ?volt-2))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var D-TAP)
                        (ctrl-loc ?loc)
                        (sen-fact ?sen-fact))
(test (or (and (< ?sen-fact 0.0) (< ?pr-set ?max-lim) (< ?volt-1 1.05))
          (and (> ?sen-fact 0.0) (> ?pr-set ?min-lim) (< ?volt-2 1.05))))
=>
     (assert (max-tap))
     (if (< ?sen-fact 0.0)
     then
     (bind ?sen-fact (* ?sen-fact -1)))
     (bind ?*max-sen* (max ?sen-fact ?*max-sen*))
     (if (= ?*max-sen* ?sen-fact)
     then
      (bind ?*max-cont* ?loc)
      (bind ?*max-type* ?type)))


 (defrule new-nearby-max-cont-tap
      (declare (salience -35))
      ?max1 <- (max-tap)
      ?control-bus <- (control-bus ? ? ?)
 =>
      (retract ?max1)
      (retract ?control-bus)
```

```
    (assert (control-bus ?*max-cont* ?*max-type* 0))
    (bind ?*max-sen* 0.0)
    (bind ?*max-cont* fyh)
    (bind ?*max-type* type))


(defrule uv-nearby-gen-voltage  ;/ This rule determines the most effective
                                ;/ usable nearby generator excit. for under
                                ;/ voltage correction
    (declare (salience -40))
    ?under-volt <- (under-volt-1 ?bus-name)
    ?control <- (control (location ?loc)
                         (ctrl-typ ?type&GV)
                         (max-lim ?max-lim)
                         (min-lim ?min-lim)
                         (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                         (qgmax ?qgmax)
                         (prior ?prior&0))
    ?qgen <- (in-state (bus-name ?loc)
                       (q-gen ?q-gen&:(< ?q-gen ?qgmax)))
    ?sensit <- (sensit (dep-var DV-LOAD)
                       (bus-name ?bus-name)
                       (ctrl-var DV-GEN)
                       (ctrl-loc ?loc)
                       (sen-fact ?sen-fact))
=>
     (assert (max-gv))
     (bind ?*max-sen* (max ?sen-fact ?*max-sen*))
     (if (= ?*max-sen* ?sen-fact)
     then
      (bind ?*max-cont* ?loc)
      (bind ?*max-type* ?type)))


(defrule ov-nearby-gen-voltage  ;/ This rule determines the most effective
                                ;/ usable nearby generator excit. for over
                                ;/ voltage correction
    (declare (salience -40))
    ?under-volt <- (over-volt ?bus-name)
    ?control <- (control (location ?loc)
                         (ctrl-typ ?type&GV)
                         (max-lim ?max-lim)
                         (min-lim ?min-lim)
                         (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                         (qgmin ?qgmin)
                         (prior ?prior&0))
    ?qgen <- (in-state (bus-name ?loc)
                       (q-gen ?q-gen&:(> ?q-gen ?qgmin)))
    ?sensit <- (sensit (dep-var DV-LOAD)
                       (bus-name ?bus-name)
                       (ctrl-var DV-GEN)
```

```
                        (ctrl-loc ?loc)
                        (sen-fact ?sen-fact))
=>
   (assert (max-gv))
   (bind ?*max-sen* (max ?sen-fact ?*max-sen*))
   (if (= ?*max-sen* ?sen-fact)
   then
    (bind ?*max-cont* ?loc)
    (bind ?*max-type* ?type)))


(defrule uqgen-nearby-gen-voltage ;/ This rule determines the most effective
                                  ;/ usable nearby generator excit. for under
                                  ;/ q-gen. correction
   (declare (salience -10))
   ?under-qgen <- (under-qgen ?bus-name)
   ?control <- (control (location ?loc&~?bus-name)
                        (ctrl-typ ?type&GV)
                        (min-lim ?min-lim)
                        (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                        (qgmin ?qgmin)
                        (prior ?prior&0))
   ?qgen <- (in-state (bus-name ?loc)
                      (q-gen ?q-gen&:(> ?q-gen ?qgmin)))
   ?sensit <- (sensit (dep-var DQ-GEN)
                      (bus-name ?bus-name)
                      (ctrl-var DV-GEN)
                      (ctrl-loc ?loc)
                      (sen-fact ?sen-fact))
=>
   (assert (max-gv))
   (bind ?*max-sen* (min ?sen-fact ?*max-sen*))
   (if (= ?*max-sen* ?sen-fact)
   then
    (bind ?*max-cont* ?loc)
    (bind ?*max-type* ?type)))


(defrule oqgen-nearby-gen-voltage ;/ This rule determines the most effective
                                  ;/ usable nearby generator excit. for over
                                  ;/ q-gen. correction
   (declare (salience -10))
   ?over-qgen <- (over-qgen ?bus-name)
   ?control <- (control (location ?loc&~?bus-name)
                        (ctrl-typ ?type&GV)
                        (max-lim ?max-lim)
                        (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                        (qgmax ?qgmax)
                        (prior ?prior&0))
   ?qgen <- (in-state (bus-name ?loc)
```

```
                         (q-gen ?q-gen&:(< ?q-gen ?qgmax)))
    ?sensit <- (sensit (dep-var DQ-GEN)
                         (bus-name ?bus-name)
                         (ctrl-var DV-GEN)
                         (ctrl-loc ?loc)
                         (sen-fact ?sen-fact))
=>
     (assert (max-gv))
     (bind ?*max-sen* (min ?sen-fact ?*max-sen*))
     (if (= ?*max-sen* ?sen-fact)
     then
      (bind ?*max-cont* ?loc)
      (bind ?*max-type* ?type)))


(defrule new-nearby-max-cont-gv
     ?max1 <- (max-gv)
     ?control-bus <- (control-bus ? ? ?)
=>
     (retract ?max1)
     (retract ?control-bus)
     (assert (control-bus ?*max-cont* ?*max-type* 0))
     (bind ?*max-sen* 0.0)
     (bind ?*max-cont* fyh)
     (bind ?*max-type* type))


(defrule under-voltage-nearby-gen               ;This rule adjusts the load
     ?update <- (update-1)                      ;bus voltage if its value
                                                ;below the limit ,using nearby
     ?under-volt-1 <- (under-volt-1 ?bus-name)  ;Gen. Voltage ,if exist
     ?control-bus <- (control-bus ?ctrl ?ctrl-typ&GV 0)
     ?voltage <- (in-state (bus-name ?bus-name) (bus-volt ?bus-volt))
     ?control <- (control (ctrl-num ?ctrl-num)
                          (location ?ctrl)
                          (ctrl-typ GV)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (ch-pr-st ?ch-pr-st)
                          (qgmax ?qgmax)
                          (prior ?prior&0))
     ?qgen <- (in-state (bus-name ?ctrl)
                        (q-gen ?q-gen))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var&DV-GEN)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-fact))
     ?sensit1 <- (sensit (dep-var DQ-GEN)
                        (bus-name ?ctrl)
```

```
                    (ctrl-var DV-GEN)
                    (ctrl-loc ?ctrl)
                    (sen-fact ?sen-fac1))
=>
   (bind ?*old-bus* ?bus-name)
   (retract ?under-volt-1)
   (retract ?control-bus)
   (bind ?d-volt (- 1.0 ?bus-volt))
   (bind ?d-sr (/ ?d-volt ?sen-fact))
   (if (<= ?d-sr (- ?max-lim ?pr-set)) ; if-1
   then
   (bind ?q-gen (+ ?q-gen (* ?d-sr ?sen-fac1)))
   (if (> ?q-gen ?qgmax)
   then
   (bind ?d-qgen (- ?qgmax ?q-gen))
   (bind ?d-srgen (/ ?d-qgen ?sen-fac1))
   (bind ?d-sr (+ ?d-sr ?d-srgen))
   (bind ?max-lim (+ ?pr-set ?d-sr))
   ) ; close 'if'
   (if (< ?d-sr 0.0001)
   then
   (modify ?control (prior 1))
   (retract ?update)
   else
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   )
   else ; of 'if-1'
   (bind ?d-sr (- ?max-lim ?pr-set))
   (bind ?q-gen (+ ?q-gen (* ?d-sr ?sen-fac1)))
   (if (> ?q-gen ?qgmax)
   then
   (bind ?d-qgen (- ?qgmax ?q-gen))
   (bind ?d-srgen (/ ?d-qgen ?sen-fac1))
   (bind ?d-sr (+ ?d-sr ?d-srgen))
   (bind ?max-lim (+ ?pr-set ?d-sr))
   ) ; close 'if'
    (if (< ?d-sr 0.0001)
   then
   (modify ?control (prior 1))
   (retract ?update)
   else
   (assert (d-chang ?d-sr))
   (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
   (assert (next-stage-1a))
   (assert (next-stage-1b))
   )
   ) ;close 'if-1'
   ) ;close the rule
```

```
(defrule over-voltage-nearby-gen                    ;This rule adjusts the load
     ?under-volt-1 <- (over-volt ?bus-name)         ;bus voltage if its value
                                                    ;above the limit ,using nearby
     ?control-bus <- (control-bus ?ctrl ?ctrl-typ&GV ?);Gen. Voltage ,if exist
     ?voltage <- (in-state (bus-name ?bus-name) (bus-volt ?bus-volt))

     ?control <- (control (ctrl-num ?ctrl-num)
                          (location ?ctrl)
                          (ctrl-typ GV)
                          (max-lim ?max-lim)
                          (min-lim ?min-lim)
                          (pr-set ?pr-set)
                          (ch-pr-st ?ch-pr-st)
                          (qgmin ?qgmin)
                          (prior ?prior&0))
     ?qgen <- (in-state (bus-name ?ctrl)
                        (q-gen ?q-gen))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var&DV-GEN)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-fact))
     ?sensit1 <- (sensit (dep-var DQ-GEN)
                        (bus-name ?ctrl)
                        (ctrl-var DV-GEN)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-fac1))
=>
      (bind ?*old-bus* ?bus-name)
      (retract ?under-volt-1)
      (retract ?control-bus)
      (bind ?d-volt (- 1.05 ?bus-volt))
      (bind ?d-sr (/ ?d-volt ?sen-fact))
      (if (>= ?d-sr (- ?min-lim ?pr-set)) ; if-1
      then
      (bind ?q-gen (+ ?q-gen (* ?d-sr ?sen-fac1)))
      (if (< ?q-gen ?qgmin)
      then
      (bind ?d-qgen (- ?qgmin ?q-gen))
      (bind ?d-srgen (/ ?d-qgen ?sen-fac1))
      (bind ?d-sr (+ ?d-sr ?d-srgen))
      (bind ?min-lim (+ ?pr-set ?d-sr))
      ) ; close 'if'
      (assert (d-chang ?d-sr))
      (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
      (assert (next-stage-1a))
      (assert (next-stage-1b))
      else ; of 'if-2'
      (bind ?d-sr (- ?min-lim ?pr-set))
      (bind ?q-gen (+ ?q-gen (* ?d-sr ?sen-fac1)))
      (if (< ?q-gen ?qgmin)
```

```
then
(bind ?d-qgen (- ?qgmin ?q-gen))
(bind ?d-srgen (/ ?d-qgen ?sen-fac1))
(bind ?d-sr (+ ?d-sr ?d-srgen))
(bind ?min-lim (+ ?pr-set ?d-sr))
) ; close 'if'
(assert (d-chang ?d-sr))
(assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
(assert (next-stage-1a))
(assert (next-stage-1b))
) ;close 'if-1'
) ;close the rule


(defrule under-qgen-local-gv      ;This rule adjusts the gen.
                                  ;bus q-gen. if its value
                                  ;below the limit ,using local
                                  ;Gen. Voltage ,if exist
    ?u-qgen <- (under-qgen ?bus-name)
    ?control-bus <- (control-bus ?bus-name ? ?)
    ?qgen <- (in-state (bus-name ?bus-name)
                       (q-gen ?q-gen))
    ?ctrl <- (control (ctrl-num ?ctrl-num)
                      (location ?bus-name)
                      (ctrl-typ ?ctrl-typ&GV)
                      (max-lim ?max-lim)
                      (min-lim ?min-lim)
                      (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                      (qgmin ?qgmin)
                      (prior ?prior&0))
    ?sensit <- (sensit (dep-var DQ-GEN)
                       (bus-name ?bus-name)
                       (ctrl-var ?ctrl-var&DV-GEN)
                       (ctrl-loc ?bus-name)
                       (sen-fact ?sen-fact))
=>
    (bind ?*old-bus* ?bus-name)
    (retract ?u-qgen)
    (retract ?control-bus)
    (bind ?d-qgen (- ?qgmin ?q-gen))
    (bind ?d-gv (/ ?d-qgen ?sen-fact))
    (if (<= ?d-gv (- ?max-lim ?pr-set))
    then
    (assert (d-chang ?d-gv))
    (assert (controller ?bus-name ?ctrl-var ?ctrl-typ))
    (assert (next-stage-1a))
    (assert (next-stage-1b))
    else
    (bind ?d-gv (- ?max-lim ?pr-set))
    (assert (d-chang ?d-gv))
    (assert (controller ?bus-name ?ctrl-var ?ctrl-typ))
    (assert (next-stage-1a))
```

```
    (assert (next-stage-1b))
    ) ; close ' if '
    ) ; close the rule


(defrule over-qgen-local-gv        ;This rule adjusts the gen.
                                   ;bus q-gen. if its value
                                   ;above the limit ,using local
                                   ;Gen. Voltage ,if exist
     ?u-qgen <- (over-qgen ?bus-name)
     ?control-bus <- (control-bus ?bus-name ? ?)
     ?qgen <- (in-state (bus-name ?bus-name)
                        (q-gen ?q-gen))
     ?ctrl <- (control (ctrl-num ?ctrl-num)
                         (location ?bus-name)
                         (ctrl-typ ?ctrl-typ&GV)
                         (max-lim ?max-lim)
                         (min-lim ?min-lim)
                         (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                         (qgmax ?qgmax)
                         (prior ?prior&0))
     ?sensit <- (sensit (dep-var DQ-GEN)
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var&DV-GEN)
                        (ctrl-loc ?bus-name)
                        (sen-fact ?sen-fact))
=>
      (bind ?*old-bus* ?bus-name)
      (retract ?u-qgen)
      (retract ?control-bus)
      (bind ?d-qgen (- ?qgmax ?q-gen))
      (bind ?d-gv (/ ?d-qgen ?sen-fact))
      (if (>= ?d-gv (- ?min-lim ?pr-set))
      then
      (assert (d-chang ?d-gv))
      (assert (controller ?bus-name ?ctrl-var ?ctrl-typ))
      (assert (next-stage-1a))
      (assert (next-stage-1b))
      else
      (bind ?d-gv (- ?min-lim ?pr-set))
      (assert (d-chang ?d-gv))
      (assert (controller ?bus-name ?ctrl-var ?ctrl-typ))
      (assert (next-stage-1a))
      (assert (next-stage-1b))
      ) ; close ' if '
        ;
      ) ; close the rule


 (defrule under-qgen-nearby-gv     ;This rule adjusts the gen.
                                   ;bus q-gen. if its value
```

123-6

```
                                ;below the limit ,using nearby
                                ;Gen. Voltage ,if exist
    ?u-qgen <- (under-qgen ?bus-name)
    ?control-bus <- (control-bus ?ctrl ?ctrl-typ&GV ?)
    ?qgen <- (in-state (bus-name ?bus-name)
                       (q-gen ?q-gen))
    ?ctrl0 <- (control (ctrl-num ?ctrl-num)
                        (location ?bus-name)
                        (ctrl-typ GV)
                        (qgmin ?qgmin))
    ?ctrl1 <- (control (ctrl-num ?ctrl-num1)
                        (location ?ctrl)
                        (ctrl-typ GV)
                        (min-lim ?min-lim)
                        (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                        (prior ?prior&0))
    ?sensit <- (sensit (dep-var DQ-GEN)
                       (bus-name ?bus-name)
                       (ctrl-var ?ctrl-var&DV-GEN)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-fact))
=>
     (bind ?*old-bus* ?bus-name)
     (retract ?u-qgen)
     (retract ?control-bus)
     (bind ?d-qgen (- ?qgmin ?q-gen))
     (bind ?d-gv (/ ?d-qgen ?sen-fact))
     (if (>= ?d-gv (- ?min-lim ?pr-set)) ; 'if-1'
     then
     (assert (d-chang ?d-gv))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     else ; of 'if-1'
     (bind ?d-gv (- ?min-lim ?pr-set))
     (assert (d-chang ?d-gv))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     ) ;close 'if-1'
     ) ;close the rule


 (defrule over-qgen-nearby-gv      ;This rule adjusts the gen.
                                   ;bus q-gen. if its value
                                   ;above the limit ,using nearby
                                   ;Gen. Voltage ,if exist
     ?u-qgen <- (over-qgen ?bus-name)
     ?control-bus <- (control-bus ?ctrl ?ctrl-typ&GV ?)
     ?qgen <- (in-state (bus-name ?bus-name)
                        (q-gen ?q-gen))
     ?ctrl0 <- (control (ctrl-num ?ctrl-num)
```

```
                          (location ?bus-name)
                          (ctrl-typ GV)
                          (qgmax ?qgmax))
    ?ctrl1 <- (control (ctrl-num ?ctrl-num1)
                          (location ?ctrl)
                          (ctrl-typ GV)
                          (max-lim ?max-lim)
                          (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                          (prior ?prior&0))
    ?sensit <- (sensit (dep-var DQ-GEN)
                       (bus-name ?bus-name)
                       (ctrl-var ?ctrl-var&DV-GEN)
                       (ctrl-loc ?ctrl)
                       (sen-fact ?sen-fact))
=>
     (bind ?*old-bus* ?bus-name)
     (retract ?u-qgen)
     (retract ?control-bus)
     (bind ?d-qgen (- ?qgmax ?q-gen))
     (bind ?d-gv (/ ?d-qgen ?sen-fact))
     (if (<= ?d-gv (- ?max-lim ?pr-set)) ; 'if-1'
     then
     (assert (d-chang ?d-gv))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     else ; of 'if-1'
     (bind ?d-gv (- ?max-lim ?pr-set))
     (assert (d-chang ?d-gv))
     (assert (controller ?ctrl ?ctrl-var ?ctrl-typ))
     (assert (next-stage-1a))
     (assert (next-stage-1b))
     ) ;close 'if-1'
     ) ;close the rule


 (defrule correction-stage-2a       ;/ This rule verifies if the propsed change
                                    ;/ in control element causes any new load bus
                                    ;/ voltage violation
     ?next-stage <- (next-stage-1a)
     ?dchange <- (d-chang ?d-ch)
     ?control <- (controller ?ctrl ?ctrl-var ?ctrl-typ)
     ?initial <- (in-state (bus-name ?bus-name)
                              (bus-type PQ) (bus-volt ?bus-volt)
                              (pr-dev ?pr-dev) (seq 0))
     ?sensit <- (sensit (dep-var DV-LOAD)
                        (bus-name ?bus-name)
                        (ctrl-var ?ctrl-var)
                        (ctrl-loc ?ctrl)
                        (sen-fact ?sen-fact))
     ?n-lbus <- (nlbus ?nlbus)
 =>
```

```
(bind ?*countt* (+ ?*countt* 1))
(bind ?d-volt (* ?d-ch ?sen-fact))
(bind ?n-volt (+ ?bus-volt ?d-volt))
(if (and (> ?n-volt 1.05) (= ?pr-dev 0.0)) ; 'if-1'
then
(bind ?new-dev (- ?n-volt 1.05))
(bind ?dd-ch (/ ?new-dev (abs ?sen-fact)))
(bind ?*max-dsr1* (max ?*max-dsr1* ?dd-ch))
(assert (not-update ?ctrl ?ctrl-var ?ctrl-typ ?d-ch))
) ; close 'if-1'
(if (and (< ?n-volt 0.95) (= ?pr-dev 0.0)) ; 'if-2'
then
(bind ?new-dev (- 0.95 ?n-volt))
(bind ?dd-ch (/ ?new-dev (abs ?sen-fact)))
(bind ?*max-dsr1* (max ?*max-dsr1* ?dd-ch))
(assert (not-update ?ctrl ?ctrl-var ?ctrl-typ ?d-ch))
) ; close 'if-2'
(modify ?initial (seq 1))
(if (> ?nlbus ?*countt*) ; 'if-3'
then
(retract ?next-stage)
(assert (next-stage-1a))
else
(retract ?next-stage)
(bind ?*countt* 0)
) ; close 'if-3'
) ; close the rule


(defrule correction-stage-2b ;/ This rule verifies if the propsed change
                              ;/ in control element causes any new gen. bus
                              ;/ q-gen. violation
   ?next-stage <- (next-stage-1b)
   ?dchange <- (d-chang ?d-ch)
   ?control <- (controller ?ctrl ?ctrl-var ?ctrl-typ)
   ?initial <- (in-state (bus-name ?bus-name)
                         (bus-type SLK|PV) (q-gen ?q-gen)
                         (pr-dev ?pr-dev) (seq ?seq&~2))
   ?ctrl0 <- (control (location ?bus-name)
                      (ctrl-typ GV)
                      (qgmin ?qgmin)
                      (qgmax ?qgmax))
   ?sensit <- (sensit (dep-var DQ-GEN)
                      (bus-name ?bus-name)
                      (ctrl-var ?ctrl-var)
                      (ctrl-loc ?ctrl)
                      (sen-fact ?sen-fact))
   ?n-gen <- (ngen ?ngen)
=>
   (bind ?*countt1* (+ ?*countt1* 1))
   (bind ?d-qgen (* ?d-ch ?sen-fact))
```

```
(bind ?n-qgen (+ ?q-gen ?d-qgen))
(if (and (> ?n-qgen ?qgmax) (= ?pr-dev 0.0)) ; 'if-1'
then
(bind ?new-dev (- ?n-qgen ?qgmax))
(bind ?dd-ch (/ ?new-dev (abs ?sen-fact)))
(bind ?*max-dsr2* (max ?*max-dsr2* ?dd-ch))
(assert (not-update ?ctrl ?ctrl-var ?ctrl-typ ?d-ch))
) ; close 'if-1'
(if (and (< ?n-qgen ?qgmin) (= ?pr-dev 0.0)) ; 'if-2'
then
(bind ?new-dev (- ?qgmin ?n-qgen))
(bind ?dd-ch (/ ?new-dev (abs ?sen-fact)))
(bind ?*max-dsr2* (max ?*max-dsr2* ?dd-ch))
(assert (not-update ?ctrl ?ctrl-var ?ctrl-typ ?d-ch))
) ; close 'if-2'
(modify ?initial (seq 2))
(if (> ?ngen ?*countt1*) ; 'if-3'
then
(retract ?next-stage)
(assert (next-stage-1b))
else
(retract ?next-stage)
(bind ?*countt1* 0)
) ; close 'if-3'
) ; close the rule


(defrule correction-stage-3 ;/ This rule & next one determine the final
                            ;/ change in the control elements without causing any
                            ;/ new violations
    (declare (salience -1))
    ?update <- (not-update ? ? ? ?)
    ?dchange <- (d-chang ?)
    ?control <- (controller ? ? ?)
=>
    (retract ?dchange)
    (retract ?control)
    (assert (max-ch =(max ?*max-dsr1* ?*max-dsr2*)))
    (bind ?*max-dsr1* 0.0)
    (bind ?*max-dsr2* 0.0)
    ) ; close the rule


(defrule correction-stage-4 ;/
    ?update-1 <- (update-1)
    ?update <- (not-update ?ctrl ?ctrl-var ?ctrl-typ ?d-ch)
    ?max-dev1 <- (max-ch ?m-ch)
    ?ctrl0 <- (control (location ?ctrl)
                       (ctrl-typ ?ctrl-typ)
                       (pr-set ?pr-set)
                       (ch-pr-st ?ch-pr-st)
```

```
                              (prior ?prior))
=>
    (retract ?update)
    (retract ?max-dev1)
    (if (> ?ch-pr-st 0.0) ; 'if-1'
    then
    (bind ?n-step (+ (integer (/ ?m-ch ?ch-pr-st)) 1))
    (bind ?m-ch (* ?n-step ?ch-pr-st))
    ) ; close 'if-1'
    (if (> ?d-ch 0.0)
    then
    (bind ?d-ch (- ?d-ch ?m-ch))
    (if (< ?d-ch 0.00001)
    then
    (modify ?ctrl0 (prior 1))
    (retract ?update-1)
    (assert (update-seq))
    else
    (bind ?pr-set (+ ?pr-set ?d-ch))
    (modify ?ctrl0 (pr-set ?pr-set) (prior 1))
    (assert (update ?ctrl ?ctrl-var ?d-ch 1)))
    else
    (bind ?d-ch (+ ?d-ch ?m-ch))
    (if (> ?d-ch -0.00001)
    then
    (modify ?ctrl0 (prior 1))
    (retract ?update-1)
    (assert (update-seq))
    else
    (bind ?pr-set (+ ?pr-set ?d-ch))
    (modify ?ctrl0 (pr-set ?pr-set) (prior 1))
    (assert (update ?ctrl ?ctrl-var ?d-ch 1)))
    ) ; close 'if'
    ) ; close the rule


(defrule correct-5a ;/ This rule & next one confirm the orginal change
                    ;/ in the selected control elements
     (declare (salience -1))
     (d-chang ?)
=>
     (if (and (= ?*max-dsr1* 0.0) (= ?*max-dsr2* 0.0)) ; 'if-4'
     then
     (assert (stage-5a) (stage-5b))
     ) ; close 'if-4'
     ) ; close the rule


(defrule correction-stage-5
     ?stage1 <- (stage-5a)
```

```
    ?stage2 <- (stage-5b)
    ?dchange <- (d-chang ?d-ch)
    ?control <- (controller ?ctrl ?ctrl-var ?ctrl-typ)
    ?ctrl0 <- (control (location ?ctrl)
                       (ctrl-typ ?ctrl-typ)
                       (pr-set ?pr-set)
                       (prior ?prior))
=>
    (retract ?stage1)
    (retract ?stage2)
    (retract ?dchange)
    (retract ?control)
    (bind ?pr-set (+ ?pr-set ?d-ch))
    (modify ?ctrl0 (pr-set ?pr-set))
    (assert (update ?ctrl ?ctrl-var ?d-ch 1))
    ) ; close the rule


(defrule update-seq
     ?update-1 <- (update-seq)
     ?voltage <- (in-state (bus-name ?bus-name)
                           (seq~0))
     (nlbus ?nlbus)
     (ngen ?ngen)
=>
     (if (>= (+ ?nlbus ?ngen) ?*count*)
     then
     (bind ?*count* (+ ?*count* 1))
     (modify ?voltage (seq 0))
     ;(retract ?update-1)
     ;(assert (update-seq))
     ) ; close 'if'
     (if (= (+ (+ ?nlbus ?ngen) 1) ?*count*)   ; 'if-3'
     then
     (bind ?*count* 1)
     (retract ?update-1)
     ) ; close 'if-3'
     ) ; close the rule


 (defrule update-voltage                          ;This rule updates the
      ?update-1 <- (update-1)                     ;voltages of load buses
                                                  ;for any change in
                                                  ;control variables

      ?update <- (update ?c-location ?c-var ?d-sr ?)

      ?voltage <- (in-state (bus-name ?bus-name) (bus-type PQ)
                            (bus-volt ?bus-volt)
                            (pr-dev ?pr-dev) (seq 1))
      ?sensit <- (sensit (dep-var DV-LOAD)
                         (bus-name ?bus-name)
```

```
                     (ctrl-var ?c-var)
                     (ctrl-loc ?c-location)
                     (sen-fact ?sen-fact))
    (not (update-qgen))
    (nlbus ?nlbus)
=>
    (if (>= ?nlbus ?*count*)
    then
    (bind ?*count* (+ ?*count* 1))
    (bind ?bus-volt (+ ?bus-volt (* ?d-sr ?sen-fact)))
    (if (> ?bus-volt 1.05)
    then
    (bind ?pr-dev (- ?bus-volt 1.05)))
    (if (< ?bus-volt 0.95)
    then
    (bind ?pr-dev (- ?bus-volt 0.95)))
    (if (and (<= ?bus-volt 1.05) (>= ?bus-volt 0.95))
    then
    (bind ?pr-dev 0.0))
    (modify ?voltage (bus-volt ?bus-volt) (pr-dev ?pr-dev) (seq 0))
    (retract ?update)
    (assert (update ?c-location ?c-var ?d-sr 1))
    ) ; close 'if'
    (if (= (+ ?nlbus 1) ?*count*)  ; 'if-3'
    then
    (bind ?*count* 1)
    (retract ?update)
    (assert (update ?c-location ?c-var ?d-sr 1))
    (assert (update-qgen))
    ) ; close 'if-3'
    ) ; close the rule


 (defrule update-qgen                                 ;This rule updates the
     ?update-1 <- (update-1)                          ;for any change in
     ?update-2 <- (update-qgen)                       ;control variables
     ?update <- (update ?c-location ?c-var ?d-sr ?)
     ;q-generation of gen. ;buses
     ?qgen <- (in-state (bus-name ?bus-name)
                        (bus-type SLK|PV)
                        (bus-volt ?bus-volt)
                        (q-gen ?q-gen)
                        (pr-dev ?pr-dev) (seq 2))
     ?ctrl0 <- (control (location ?bus-name)
                        (ctrl-typ GV)
                        (qgmin ?qgmin)
                        (qgmax ?qgmax))
     ?sensit <- (sensit (dep-var DQ-GEN)
                        (bus-name ?bus-name)
                        (ctrl-var ?c-var)
                        (ctrl-loc ?c-location)
                        (sen-fact ?sen-fact))
```

```
    (ngen ?ngen)
=>
    (if (>= ?ngen ?*count*)
    then
    (bind ?*count* (+ ?*count* 1))
    (bind ?q-gen (+ ?q-gen (* ?d-sr ?sen-fact)))
    (if (< ?q-gen ?qgmin)
    then
    (bind ?pr-dev (- ?q-gen ?qgmin)))
    (if (> ?q-gen ?qgmax)
    then
    (bind ?pr-dev (- ?q-gen ?qgmax)))
    (if (and (>= ?q-gen ?qgmin) (<= ?q-gen ?qgmax))
    then
    (if (< (- ?qgmax ?q-gen) 0.00000000001)
    then
    (bind ?q-gen (round ?q-gen)))
    (if (< (- ?q-gen ?qgmin) 0.00000000001)
    then
    (bind ?q-gen (round ?q-gen)))
    (bind ?pr-dev 0.0))
    (if (eq ?bus-name ?c-location)
    then
    (bind ?bus-volt (+ ?bus-volt ?d-sr)))
    (modify ?qgen (bus-volt ?bus-volt) (q-gen ?q-gen) (pr-dev ?pr-dev) (seq 0))
    ) ; close 'if'
    (if (= (+ ?ngen 1) ?*count*) ; 'if-1'
    then
    (bind ?*count* 1)
    (retract ?update)
    (retract ?update-1)
    (retract ?update-2)
    (assert (update-3))
    else
    (retract ?update)
    (assert (update ?c-location ?c-var ?d-sr 1))
    ) ; close 'if-1'
    ) ; close the rule


(defrule acticate-no-controller ;/ This rule verifies if there is no
                                ;/ adequate control measures
    (declare (salience -65))
    (and
    (not (max-under-volt))
    (not (max-over-volt))
    (not (max-under-qgen))
    (not (max-over-qgen))
    (not (update-1))
    ) ; close 'and'
    ?initial <- (in-state (bus-name ?bus-name)
                          (pr-dev ?pr-dev&~0.0))
```

```
=>
     (assert (update-1)))

(defrule sucess-ful
     (declare (salience -66))
     (update-1)
     ?update <- (update-3)
=>
     (retract ?update))



(defrule open-print  ;/ This rule open the output file
     (declare (salience -69))
     (or
     (update-1)
     (update-3)
     )
=>
     (open "voltage.out" volt "w")
     (assert (file-opened)))


(defrule print-new-voltages1 ;/ The following rule prints the resuls
     (declare (salience -70))
     (file-opened)
=>
  (printout volt "THE EXPERT SYSTEM RESULTS FOR VOLTAGE/VAR ADJUSTMENT"
  crlf)
  (printout volt " " crlf))


(defrule print-new-voltages2
     (declare (salience -71))
     ?voltage <- (in-state (bus-name ?bus-name)
                           (bus-type PQ)
                           (bus-volt ?bus-volt))
     (file-opened)
=>
     (if (= ?*countt* 0)
     then
  (printout volt "The New Voltages of the Load Buses are as follow:-" crlf)
  (printout volt " " crlf)
   (bind ?*countt* 1)
   ) ; close 'if'
  (format volt "BUS: %-3d%-13s%-10s%-2.4f%n"
   ?bus-name "  TYPE: PQ  " " VOLTAGE=  " ?bus-volt))


(defrule print-new-qgen2
   (declare (salience -75))
```

```
 (file-opened)
 ?qgen <- (in-state (bus-name ?bus-name)
                    (bus-type ?type&:(or (eq ?type SLK)
                                         (eq ?type PV)))
                    (q-gen ?q-gen))
=>
  (if (= ?*countt1* 0)
  then
 (printout volt " " crlf)
 (printout volt "The New Q-Gen. of the Gen. Buses are as follow:-" crlf)
 (printout volt " " crlf)
  (bind ?*countt1* 1)
  )  ; close 'if'
 (format volt "BUS: %-3d%-8s%-3s%-10s%-4.4f%n"
  ?bus-name "  TYPE: " ?type " Q-GEN.= " ?q-gen))


(defrule print-new-control-settings
  (declare (salience -80))
  (file-opened)
  ?control <- (control (ctrl-num ?ctrl-num)
                       (location ?location)
                       (ctrl-typ ?ctrl-typ)
                       (pr-set ?pr-set))
=>
  (if (= ?*countt1* 1)
  then
 (printout volt " " crlf)
 (printout volt "The New Settings of Controllers are as follow:-" crlf)
 (printout volt " " crlf)
  (bind ?*countt1* 2)
  )  ; close 'if'
  (printout volt " " crlf)
  (printout volt "CONT. SEQ. # " ?ctrl-num "  LOCATION :  " ?location crlf)
  (printout volt  "CONT. TYPE :  " ?ctrl-typ  crlf)
  (format volt  "  PRES. SETTING :%-4.6f%n" ?pr-set))


(defrule print-no-controller
  (declare (salience -85))
  (update-1)
  (file-opened)
=>
  (printout volt " " crlf)
  (printout volt " NO ADEQUTE CONTROLLER AVAILABLE. " crlf))


(defrule close-print ;/ This rule closes the output file
  (declare (salience -90))
  (file-opened)
=>
  (close volt))
```

APPENDIX - B.2

COMPUTER LISTING FOR KB OF

CORRECTIVE GENERATION

RE-SCHEULING PROCESS

KNOWLEDGE BASE FOR GENERATION RE-SCHEDULING

```
(defglobal ;/ GLOBAL VARIABLES
     ?*ngen* = 0
     ?*nline* = 0
     ?*max-sen* = 0.0
     ?*min-sen* = 0.0
     ?*plus-cont* = fyh1
     ?*min-cont* = fyh2
     ?*dgen-plus* = 0.0
     ?*dgen-min* = 0.0
     ?*o-line* = fyh
     ?*pf-line* = 0.0
     ?*max-dev* = 0.0
     ?*old-line* = fyh
     ?*count* = 1
     ?*max-dgen* = 0.0
     ?*max-dgen1* = 0.0)


(deftemplate generate ;/ DATA BASE FOR GENERATORS
     (slot gen-num (default ?NONE))
     (slot gen-loc (type SYMBOL) (default ?NONE))
     (slot max-lim (default ?NONE))
     (slot min-lim (default ?NONE))
     (slot pr-set (default ?NONE))
     (slot prior (default ?NONE)))


(deftemplate lsensit ;/ DATA BASE FOR SENSITIVITY COFFICIENTS
     (slot line-num (default ?NONE))
     (slot gen-loc (default ?NONE))
     (slot sen-fact (default ?NONE)))


(deftemplate pflow ;/ DATA BASE FOR LINE POWER FLOWS
     (slot line-num (default ?NONE))
     (slot bus-a (type SYMBOL) (default ?NONE))
     (slot bus-b (type SYMBOL) (default ?NONE))
     (slot pflow (default ?NONE))
     (slot max-flow (default ?NONE))
     (slot pr-dev (type NUMBER) (default 0.0))
     (slot seq (type NUMBER) (default 0))
     (slot priorty (type NUMBER) (default 0)))


(defrule open-file-pflow ;/ THIS RULE OPEN THE FILE CONTAINING INITIAL
     (initial-fact)      ;/ VALUES OF LINE POWER FLOWS OBTAINED FROM L.F.
```

```
=>
    (open "pflow.dat" pflow "r")
    (assert (initial-fact2))
    (assert (read-file-pflow)))


(defrule read-file-pflow ;/ THIS RULE READS THE INITIAL VALUES OF LINE
                         ;/ FLOWS FROM A FILE
    ?read-pflow <- (read-file-pflow)
 =>
    (retract ?read-pflow)
    (bind ?line-num (read pflow))
    (bind ?bus-a (read pflow))
    (bind ?bus-b (read pflow))
    (bind ?pflow (read pflow))
    (bind ?max-flow (read pflow))
    (if (neq ?line-num EOF-PFLOW) ; 'if-1'
    then
    (bind ?pr-dev (- (abs ?pflow) ?max-flow))
    (if (<= ?pr-dev 0.0)
    then
    (bind ?pr-dev 0.0))
    (bind ?*nline* (+ ?*nline* 1))
    (assert (pflow (line-num ?line-num) (bus-a ?bus-a)
                   (bus-b ?bus-b) (pflow ?pflow) (max-flow ?max-flow)
                   (pr-dev ?pr-dev)))
    (assert (read-file-pflow))
    ) ; close ' if-1 '
    (if (eq ?line-num EOF-PFLOW) ; 'if-2'
    then
    (assert (EOF-PFLOW))
    (close pflow) ; close the file
    ) ; close ' if-2 '
    ) ; close the rule


 (defrule pflow-violation-check ;/ THIS RULE CHECKS THE LINE OVERLOADS
    ?initial <- (initial-fact)
    ?pflow-1 <- (pflow (line-num ?line-num) (pflow ?pflow)
                       (max-flow ?max-flow))
    (test (> (abs ?pflow) ?max-flow))
 =>
    (retract ?initial)
    (assert (violation-exist)))


 (defrule print-no-violation ;/ THIS RULE PRINTS " NO OVERLOADS " MESSAGE
    (declare (salience -5)) ;/ IN CASE NO OVERLOAD EXIST
    (not (violation-exist))
 =>
    (printout t " THERE IN NO OVER-LOADED LINE " crlf))
```

```
(defrule open-file-violation ;/ THIS RULE OPEN GENERATOR & SENSITIVITY
    (violation-exist)         ;/ DATA FILES IN  CASE OVERLOAD EXIST
=>
    (open "gen.dat" gen "r")
    (open "lsensit.dat" lsensit "r")
    (assert (read-file-gen)
            (read-file-lsensit)))


(defrule read-file-gen  ;/ READS THE GENERATOR DATA FROM A FILE AND
    ?read-gen <- (read-file-gen) ;/ SAVE IT IN A DATA BASE
=>
    (retract ?read-gen)
    (bind ?gen-num (read gen))
    (bind ?gen-loc (read gen))
    (bind ?max-lim (read gen))
    (bind ?min-lim (read gen))
    (bind ?pr-set (read gen))
    (if (neq ?gen-num EOF-GEN) ; 'if-1'
    then
    (bind ?*ngen* (+ ?*ngen* 1))
    (assert (generate (gen-num ?gen-num) (gen-loc ?gen-loc)
                (max-lim ?max-lim) (min-lim ?min-lim)
                (pr-set ?pr-set) (prior 1)))
    (assert (read-file-gen))
    ) ; close ' if-1 '
    (if (eq ?gen-num EOF-GEN) ; 'if-2'
    then
    (assert (EOF-GEN))
    (close gen) ; close the file
    ) ; close ' if-2 '
    ) ; close the rule


(defrule read-file-lsensit ;/ READS THE SENSITIVITY DATA FROM A FILE AND
                           ;/ SAVE IT IN A DATA BASE
    ?read-lsensit <- (read-file-lsensit)
=>
    (retract ?read-lsensit)
    (bind ?line-num (read lsensit))
    (bind ?gen-loc (read lsensit))
    (bind ?sen-fact (read lsensit))
    (if (neq ?line-num EOF-LSENSIT) ; 'if-1'
    then
    (assert (lsensit (line-num ?line-num) (gen-loc ?gen-loc)
                (sen-fact ?sen-fact)))
    (assert (read-file-lsensit))
    ) ; close ' if-1 '
    (if (eq ?line-num EOF-LSENSIT) ; 'if-2'
    then
    (assert (EOF-LSENSIT))
```

```
    (close lsensit) ; close the file
    ) ; close ' if-2 '
    ) ; close the rule


(defrule over-load-check-1 ;/ THIS RULE & NEXT ONE IDENTIFY THE MOST
     (EOF-PFLOW)              ;/ OVERLOADED LINE
     (EOF-GEN)
     (EOF-LSENSIT)
     ?pflow-1 <- (pflow (line-num ?line-num)
                        (pflow ?pflow) (pr-dev ?pr-dev)
                        (priorty ?priorty&0))
     (test (> ?pr-dev 0.0))
     (not (update))
     (not (last-round))
=>
     (if (eq ?*old-line* ?line-num)  ; 'if-1'
     then
     (assert (max-over-load))
     (assert (update))
     (bind ?*pf-line* ?pflow)
     (bind ?*o-line* ?line-num)
     else
     (assert (max-over-load))
     (bind ?*max-dev* (max ?*max-dev* ?pr-dev))
     (if (= ?*max-dev* ?pr-dev)
     then
     (bind ?*pf-line* ?pflow)
     (bind ?*o-line* ?line-num))
     ) ; close 'if-1'
     ) ; close the rule



(defrule over-load-check-2
     (declare (salience -5))
     ?maxx <- (max-over-load)
=>
     (retract ?maxx)
     (bind ?line-num ?*o-line*)
     (bind ?pflow ?*pf-line*)
     (bind ?*o-line* fyh)
     (bind ?*pf-line* 0.0)
     (bind ?*max-dev* 0.0)
     (assert (over-load-line ?line-num ?pflow)
             (gen-max-sen-plus FYH1)
             (gen-max-sen-minus FYH2)
             (update)))


(defrule new-line-check ;/ THIS RULE CHECKS FOR NEW OVERLOADED LINE
     (declare (salience 5))
```

```
    ?over-load <- (over-load-line ?line-num ?)
    ?gen <- (generate (gen-loc ?gen-loc) (prior ?prior&~0))

=>
    (if (neq ?line-num ?*old-line*)
    then
    (modify ?gen (prior 0))
    ) ; close if
    ) ; close the rule


(defrule change-line-priority ;/ THIS RULE CHANGE THE OLD OVERLOADED
    (declare (salience -20))  ;/ LINE IF THE OVERLOAD CAN NOT BE ELIMINATED
    ?over-load <- (over-load-line ?line-num ?)
    ?gen-plus <- (gen-max-sen-plus ?)
    ?gen-minus <- (gen-max-sen-minus ?)
    ?update <- (update)
    ?pflow-1 <- (pflow (line-num ?line-num)
                       (priorty ?priorty))
=>
    (retract ?over-load)
    (retract ?gen-plus)
    (retract ?gen-minus)
    (retract ?update)
    (modify ?pflow-1 (priorty 1))
    ) ; close the rule


(defrule max-gen-minus-plus-flow ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR
                                 ;/ AND HAS MARGIN FOR DECREASE IN GEN. FOR
                                 ;/ POSITIVE POWER FLOW
    ?over-load <- (over-load-line ?line-num ?pflow&:(> ?pflow 0.0))
    ?gen <- (generate (gen-loc ?gen-loc) (min-lim ?min-lim)
                      (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                      (prior ?prior&0))
    ?lsensit1 <- (lsensit (line-num ?line-num) (gen-loc ?gen-loc)
                          (sen-fact ?sen-fact&:(> ?sen-fact 0.0)))
=>
    (bind ?*old-line* ?line-num)
    (assert (max-gen minus))
    (bind ?*max-sen* (max ?*max-sen* ?sen-fact))
    (if (= ?*max-sen* ?sen-fact)
    then
    (bind ?*min-cont* ?gen-loc)))


(defrule max-gen-plus-plus-flow ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR
                                ;/ AND HAS MARGIN FOR INCREASE IN GEN. FOR
                                ;/ POSITIVE POWER FLOW
    ?over-load <- (over-load-line ?line-num ?pflow&:(> ?pflow 0.0))
```

```
    ?gen <- (generate (gen-loc ?gen-loc) (max-lim ?max-lim)
                      (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                      (prior ?prior&0))
    ?lsensit1 <- (lsensit (line-num ?line-num) (gen-loc ?gen-loc)
                          (sen-fact ?sen-fact&:(< ?sen-fact 0.0)))
=>
    (bind ?*old-line* ?line-num)
    (assert (max-gen plus))
    (bind ?*min-sen* (min ?*min-sen* ?sen-fact))
    (if (= ?*min-sen* ?sen-fact)
    then
    (bind ?*plus-cont* ?gen-loc)))


(defrule max-gen-minus-minus-flow ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR
                                  ;/ AND HAS MARGIN FOR DECREASE IN GEN. FOR
                                  ;/ NEGATIVE POWER FLOW
    ?over-load <- (over-load-line ?line-num ?pflow&:(< ?pflow 0.0))
    ?gen <- (generate (gen-loc ?gen-loc) (min-lim ?min-lim)
                      (pr-set ?pr-set&:(> ?pr-set ?min-lim))
                      (prior ?prior&0))
    ?lsensit1 <- (lsensit (line-num ?line-num) (gen-loc ?gen-loc)
                          (sen-fact ?sen-fact&:(< ?sen-fact 0.0)))
=>
    (bind ?*old-line* ?line-num)
    (assert (max-gen minus))
    (bind ?*max-sen* (min ?*max-sen* ?sen-fact))
    (if (= ?*max-sen* ?sen-fact)
    then
    (bind ?*min-cont* ?gen-loc)))


(defrule max-gen-plus-minus-flow  ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR
                                  ;/ AND HAS MARGIN FOR INCREASE IN GEN. FOR
                                  ;/ NEGATIVE POWER FLOW
    ?over-load <- (over-load-line ?line-num ?pflow&:(< ?pflow 0.0))
    ?gen <- (generate (gen-loc ?gen-loc) (max-lim ?max-lim)
                      (pr-set ?pr-set&:(< ?pr-set ?max-lim))
                      (prior ?prior&0))
    ?lsensit1 <- (lsensit (line-num ?line-num) (gen-loc ?gen-loc)
                          (sen-fact ?sen-fact&:(> ?sen-fact 0.0)))
=>
    (bind ?*old-line* ?line-num)
    (assert (max-gen plus))
    (bind ?*min-sen* (max ?*min-sen* ?sen-fact))
    (if (= ?*min-sen* ?sen-fact)
    then
    (bind ?*plus-cont* ?gen-loc)))


(defrule nearby-gen ;/ STORES THE MOST EFFECTIVE GENERATORS WITH PLUS
                    ;/ MARGIN AND MINUS MARGIN
```

```
    (declare (salience -10))
    ?max-gen <- (max-gen ?max)
    ?gen-plus <- (gen-max-sen-plus ?)
    ?gen-minus <- (gen-max-sen-minus ?)
=>
    (if (eq ?max plus)
    then
    (retract ?gen-plus)
    (retract ?max-gen)
    (assert (gen-max-sen-plus ?*plus-cont*))
    (bind ?*min-sen* 0.0)
    (bind ?*plus-cont* FYH1)
    ) ; close ' if '
    (if (eq ?max minus)
    then
    (retract ?gen-minus)
    (retract ?max-gen)
    (assert (gen-max-sen-minus ?*min-cont*))
    (bind ?*max-sen* 0.0)
    (bind ?*min-cont* FYH2)
    ) ; close ' if '
    ) ; close of the rule


(defrule gen-reschedule-plus-minus ;/ DETERMINES THE MOST EFFECTIVE GENERATOR
                                   ;/ EITHER PLUS OR MINUS
    (declare (salience -15))
    ?over-load <- (over-load-line ?line-num ?)
    ?gen-plus-1 <- (gen-max-sen-plus ?gen-plus&~FYH1)
    ?gen-minus-1 <- (gen-max-sen-minus ?gen-min&~FYH2)
    ?pflow1 <- (pflow (line-num ?line-num) (pflow ?pflow)
                      (max-flow ?max-flow) (pr-dev ?pr-dev))
    ?lsens1a <- (lsensit (line-num ?line-num)
                         (gen-loc ?gen-plus) (sen-fact ?sen1a))
    ?lsens1b <- (lsensit (line-num ?line-num)
                         (gen-loc ?gen-min) (sen-fact ?sen1b))
    ?gen1 <- (generate (gen-loc ?gen-plus) (max-lim ?max-lim1)
                       (pr-set ?pr-set1))
    ?gen2 <- (generate (gen-loc ?gen-min) (min-lim ?min-lim2)
                       (pr-set ?pr-set2))
=>
    (retract ?over-load)
    (retract ?gen-plus-1)
    (retract ?gen-minus-1)
    (bind ?max-gen (max (abs ?sen1a) (abs ?sen1b)))
    (if (> ?pflow 0.0) ; 'if-1'
    then
    (bind ?d-flow (- ?max-flow ?pflow))
    else
    (bind ?d-flow (- (abs ?pflow) ?max-flow)))
    (if (eq ?max-gen (abs ?sen1a)) ; 'if-2'
    then
```

123-6

```
(bind ?d-gen1 (/ ?d-flow ?sen1a))
(bind ?gen-av1 (- ?max-lim1 ?pr-set1))
(if (<= ?d-gen1 ?gen-av1)~
then
(assert (gen-plus ?gen-plus ?d-gen1)
        (count 1))
(assert (next-stage))
else
(assert (gen-plus ?gen-plus ?gen-av1)
        (count 1))
(assert (next-stage))
)   ; 'if'
else  ; 'if-2'
(bind ?d-gen2 (/ ?d-flow ?sen1b))
(bind ?gen-av2 (- ?min-lim2 ?pr-set2))
(if (>= ?d-gen2 ?gen-av2)
then
(assert (gen-minus ?gen-min ?d-gen2)
        (count 1))
(assert (next-stage))
else
(assert (gen-minus ?gen-min ?gen-av2)
        (count 1))
(assert (next-stage))
)   ; 'if'
) ; 'close 'if-2'
) ; close the rule


(defrule gen-reschedule-plus ;/ DETERMINES THE AMOUNT OF INCREASE IN
                              ;/ GENERATION FOR MOST OVERLOADED LINE
   (declare (salience -15))
   ?over-load <- (over-load-line ?line-num ?)
   ?gen-plus-1 <- (gen-max-sen-plus ?gen-plus&~FYH1)
   ?gen-minus-1 <- (gen-max-sen-minus ?gen-min&FYH2)
   ?pflow1 <- (pflow (line-num ?line-num) (pflow ?pflow)
                     (max-flow ?max-flow) (pr-dev ?pr-dev))
   ?lsens1 <- (lsensit (line-num ?line-num)
                       (gen-loc ?gen-plus) (sen-fact ?sen1))
   ?gen1 <- (generate (gen-loc ?gen-plus) (max-lim ?max-lim)
                      (pr-set ?pr-set))
=>
   (retract ?over-load)
   (retract ?gen-plus-1)
   (retract ?gen-minus-1)
   (if (> ?pflow 0.0)
   then
   (bind ?d-flow (- ?max-flow ?pflow))
   else
   (bind ?d-flow (- (abs ?pflow) ?max-flow)))
   (bind ?d-gen1 (/ ?d-flow ?sen1))
   (bind ?gen-av1 (- ?max-lim ?pr-set))
```

```
(if (<= ?d-gen1 ?gen-av1)
then
(assert (gen-plus ?gen-plus ?d-gen1)
        (count 1))
(assert (next-stage))
else
(assert (gen-plus  ?gen-plus ?gen-av1)
        (count 1))
(assert (next-stage))
) ; close ' if '
) ; close the rule


(defrule gen-reschedule-minus      ;/ DETERMINES THE AMOUNT OF DECREASE IN
                                   ;/ GENERATION FOR MOST OVERLOADED LINE
  (declare (salience -15))
  ?over-load <- (over-load-line ?line-num ?)
  ?gen-plus-1 <- (gen-max-sen-plus ?gen-plus&FYH1)
  ?gen-minus-1 <- (gen-max-sen-minus ?gen-min&~FYH2)
  ?pflow1 <- (pflow (line-num ?line-num) (pflow ?pflow)
                    (max-flow ?max-flow) (pr-dev ?pr-dev))
  ?lsens1 <- (lsensit (line-num ?line-num)
                      (gen-loc ?gen-min) (sen-fact ?sen1))
  ?gen1 <- (generate (gen-loc ?gen-min) (min-lim ?min-lim)
                     (pr-set ?pr-set))
=>
  (retract ?over-load)
  (retract ?gen-plus-1)
  (retract ?gen-minus-1)
  (if (> ?pflow 0.0)
  then
  (bind ?d-flow (- ?max-flow ?pflow))
  else
  (bind ?d-flow (- (abs ?pflow) ?max-flow)))
  (bind ?d-gen1 (/ ?d-flow ?sen1))
  (bind ?gen-av1 (- ?min-lim ?pr-set))
  (if (>= ?d-gen1 ?gen-av1)
  then
  (assert (gen-minus  ?gen-min ?d-gen1)
          (count 1))
  (assert (next-stage))
  else
  (assert (gen-minus  ?gen-min ?gen-av1)
          (count 1))
  (assert (next-stage))
  ) ; close ' if '
  ) ; close the rule


(defrule gen-reschedule-stage2 ;/ THIS RULE VERIFIES IF THE PROPOSED CHANGE
                               ;/ IN GENERATION CAUSES ANY NEW OVERLOADS
  ?next-stage <- (next-stage)
```

```
 (or
(gen-plus ?gen ?fdgen1)
(gen-minus ?gen ?fdgen1)
 ) ; close of 'or'
?pflow1 <- (pflow (line-num ?line-num)
                  (pflow ?pflow)
                  (max-flow ?max-flow) (pr-dev ?pr-dev) (seq 0))
?lsensit1 <- (lsensit (line-num ?line-num)
                      (gen-loc ?gen) (sen-fact ?sen1))
?count1 <- (count ?count)
=>
 (bind ?count (+ ?count 1))
 (bind ?d-pflow (* ?fdgen1 ?sen1))
 (bind ?n-pflow (+ ?pflow ?d-pflow))
 (bind ?new-dev (- (abs ?n-pflow) ?max-flow))
 (if (and (> ?new-dev 0.0) (= ?pr-dev 0.0)) ; ' if-1'
 then
 (bind ?ch-dev (- ?new-dev ?pr-dev))
 (if (> ?ch-dev 0.0) ; ' if-2'
 then
 (bind ?dgen (/ ?ch-dev (abs ?sen1)))
 (bind ?*max-dgen* (max ?*max-dgen* ?dgen))
 (assert (not-update ?gen ?fdgen1))
 ) ; close ' if-2'
 ) ; close ' if-1'
 (modify ?pflow1 (seq 1))
 (if (<= ?count ?*nline*) ; 'if-4'
 then
 (retract ?count1)
 (assert (count ?count))
 else
 (retract ?next-stage)
 (retract ?count1)
 (assert (count 1))
 (if (= ?*max-dgen* 0.0)
 then
 (assert (stage-5))
 else
 (assert (stage-3)))
 ) ; close 'if-4'
 ) ; close the rule


(defrule gen-reschdule-stage3 ;/ THIS RULE & NEXT ONE DETERMINE THE FINAL
                              ;/ CHANGE IN GENERATION WITHOUT CAUSIN NEW
                              ;/ OVERLOADS
  ?stage <- (stage-3)
  ?update <- (not-update ?gen ?fdgen1)
  (or
  ?gen1 <- (gen-plus ? ?)
  ?gen1 <- (gen-minus ? ?)
```

```
    ) ; close of 'or'
=>
     (retract ?stage)
     (retract ?gen1)
     (assert (max-dgen ?*max-dgen*))
     (bind ?*max-dgen1* 0.0)
     (bind ?*max-dgen* 0.0))


(defrule gen-reschdule-stage4
     ?update-1 <- (update)
     ?count <- (count ?)
     ?update <- (not-update ?gen ?fdgen)
     ?max-dev1 <- (max-dgen ?m-dgen)
     ?gen1 <- (generate (gen-loc ?gen) (max-lim ?max-lim)
                        (min-lim ?min-lim) (pr-set ?pr-set))
=>
     (retract ?update)
     (retract ?max-dev1)
     (if (> ?fdgen 0.0)
     then
     (bind ?fdgen (- ?fdgen ?m-dgen))
     (if (< ?fdgen 0.0000001)
     then
     (modify ?gen1 (prior 1))
     (retract ?update-1)
     (retract ?count)
     (assert (update-seq))
     else
     (bind ?pr-set (+ ?pr-set ?fdgen))
     (modify ?gen1 (pr-set ?pr-set) (prior 1))
     (assert (update ?gen ?fdgen)))
     else
     (bind ?fdgen (+ ?fdgen ?m-dgen))
     (if (> ?fdgen -0.0000001)
     then
     (modify ?gen1 (prior 1))
     (retract ?update-1)
     (retract ?count)
     (assert (update-seq))
     else
     (bind ?pr-set (+ ?pr-set ?fdgen))
     (modify ?gen1 (pr-set ?pr-set) (prior 1))
     (assert (update ?gen ?fdgen)))
     ) ; close 'if'

     ) ; close the rule


  (defrule update-seq
     ?update-1 <- (update-seq)
```

```
     ?pflow <- (pflow (line-num ?line-num) (seq~0))
=>
     (if (>= ?*nline* ?*count*)
     then
     (bind ?*count* (+ ?*count* 1))
     (modify ?pflow (seq 0))
     ) ; close 'if'
     (if (= (+ ?*nline* 1) ?*count*)  ; 'if-3'
     then
     (bind ?*count* 1)
     (retract ?update-1)
     ) ; close 'if-3'
     ) ; close the rule


(defrule gen-reschedule-stage5
     ?stage <- (stage-5)
     (or
     ?gen1 <- (gen-plus ?gen ?fdgen1)
     ?gen1 <- (gen-minus ?gen ?fdgen1)
     ) ; close of 'or'
     ?gen0 <- (generate (gen-loc ?gen) (max-lim ?max-lim)
                        (min-lim ?min-lim) (pr-set ?pr-set))
=>
     (retract ?stage)
     (retract ?gen1)
     (assert (update ?gen ?fdgen1))
     (bind ?pr-set (+ ?pr-set ?fdgen1))
     (modify ?gen0 (pr-set ?pr-set))
     ) ; close the rule


(defrule update-pflow-plus-or-minus ;/ THIS RULE UPDATES THE POWER FLOWS
                                    ;/ OF THE LINES FOR ANY CHANGE IN GENERATION
     ?update <- (update)
     ?update-1 <- (update ?gen ?fdgen1)
     ?pflow1 <- (pflow (line-num ?line-num)
                       (pflow ?pflow)
                       (max-flow ?max-flow) (pr-dev ?pr-dev))
     ?lsensit1 <- (lsensit (line-num ?line-num)
                           (gen-loc ?gen) (sen-fact ?sen1))
     ?count1 <- (count ?count)
=>
     (bind ?count (+ ?count 1))
     (bind ?d-pflow (* ?fdgen1 ?sen1))
     (bind ?pflow (+ ?pflow ?d-pflow))
     (bind ?pr-dev (- (abs ?pflow) ?max-flow))
     (if (<= ?pr-dev 0.0000000001) ; ' if-2'
     then
     (bind ?pr-dev 0.0)
     ) ; close ' if-2'
```

```
    (modify ?pflow1 (pflow ?pflow) (pr-dev ?pr-dev) (seq 0))
    (if (<= ?count ?*nline*) ; 'if-3'
    then
    (retract ?count1)
    (assert (count ?count))
    else
    (retract ?update)
    (retract ?update-1)
    (retract ?count1)
    (assert (update-3))
    ) ; close 'if-3'
    ) ; close the rule


(defrule last-round ;/ THIS RULE & NEXT RULES RESET THE OVERLOADED LINES
                    ;/ FOR LAST ROUND OF CORRECTION PROCCESS
     (declare (salience -21))
     (initial-fact2)
     ?pflow-1 <- (pflow (line-num ?line-num)
                        (pr-dev ?pr-dev&:(> ?pr-dev 0.0))
                        (priorty ?priorty&1))
=>
     (assert (last-round))
     (bind ?*old-line* fyh)
     (modify ?pflow-1 (priorty 0)))


(defrule confirm-last-round
     (declare (salience -22))
     ?last <- (last-round)
     ?inital <- (initial-fact2)
=>
     (retract ?inital)
     (retract ?last)
     (assert (last-round-ok)))


(defrule activate-no-controller ;/ THIS RULE VERIFIES IF THERE IS NO
                                ;/ ADEQUATE GENERATION SHIFT
     (declare (salience -23))
     ?pflow-1 <- (pflow (line-num ?line-num)
                        (pr-dev ?pr-dev&:(> ?pr-dev 0.0)))
=>
     (assert (update)))


(defrule sucess-ful
     (declare (salience -24))
     (update)
     ?update <- (update-3)
```

```
=>
    (retract ?update))


(defrule open-print ;/ THIS RULE OPEN THE OUTPUT FILE
    (declare (salience -26))
    (or
    (update-3)
    (update))
=>
    (open "generl.out" generl "w"))


(defrule print-new-pflow1 ;/ THE FOLLOWING RULES PRINT THE FINAL RESULTS
    (declare (salience -28))
    (update-3)
=>
    (printout generl "EXPERT SYSTEM RESULTS FOR GENERATION RESCHEDULING"
    crlf)
    (printout generl " " crlf)
    (printout generl "The New Values of the Power Flows are as follow:-" crlf
     " " crlf))


(defrule print-new-pflow2
    (declare (salience -30))
    (update-3)
    ?pflowl <- (pflow (line-num ?line-num) (bus-a ?bus-a)
                      (bus-b ?bus-b) (pflow ?pflow)
                      (max-flow ?max-flow) (pr-dev ?pr-dev))
=>
(format generl "LINE NO. :%-2d%-10s%-3d%-3s%-3d%-10s%-4.4f%-16s%-4.4f%n"
 ?line-num " BETWEEN : " ?bus-a " - "
?bus-b  "  P. FLOW : " ?pflow "  P. FLOW LIM. : " ?max-flow))


(defrule print-gen-title
    (declare (salience -34))
    (update-3)
=>
    (printout generl "  " crlf)
    (printout generl "The New Generation Schedule is as follow:-" crlf))


(defrule print-new-gener
    (declare (salience -35))
    (update-3)
    ?gen <- (generate (gen-loc ?gen-loc) (max-lim ?max-lim)
```

```
                        (min-lim ?min-lim) (pr-set ?pr-set))
=>
 (printout generl " " crlf)
 (format generl " GEN. LOCAT. :%-3d%-10s%-4.4f%n"
 ?gen-loc " PR. SET : " ?pr-set ))


(defrule print-no-controller1
     (declare (salience -38))
     (update)
=>
     (printout generl "EXPERT SYSTEM RESULTS FOR GENERATION RESCHEDULING"
     crlf)
     (printout generl " " crlf)
     (printout generl " NO ADEQUTE CONTROLLER AVAILABLE. " crlf)
     (printout generl " " crlf))


(defrule print-no-controller2
     (declare (salience -40))
     (update)
     ?pflowl <- (pflow (line-num ?line-num) (bus-a ?bus-a)
                       (bus-b ?bus-b) (pflow ?pflow)
                       (max-flow ?max-flow) (pr-dev ?pr-dev))
=>
     (printout generl " " crlf)
     (printout generl " " crlf)
(format generl "LINE NO. :%-2d%-10s%-3d%-3s%-3d%-10s%-4.4f%-16s%-4.4f%n"
?line-num " BETWEEN : " ?bus-a " - "
?bus-b  "  P. FLOW : " ?pflow "  P. FLOW LIM. : " ?max-flow))


(defrule print-no-controller3
     (declare (salience -42))
     (update)
     ?gen <- (generate (gen-loc ?gen-loc) (max-lim ?max-lim)
                       (min-lim ?min-lim) (pr-set ?pr-set))
=>
     (printout generl " " crlf)
     (printout generl " " crlf)
  (format generl " GEN. LOCAT. :%-3d%-10s%-4.4f%n"
  ?gen-loc " PR. SET : " ?pr-set ))


(defrule close-print ;/ THIS RULE CLOSES THE OUTPUT FILE
     (declare (salience -50))
     (or
     (update-3)
     (update))
=>
     (close generl))
```

APPENDIX - B.3

COMPUTER LISTING FOR KB OF

VOLTAGE ANGLE ADJUSTMENT PROCESS

KNOWLEDGE BASE FOR VOLTAGE ANGLE ADJUSMENT

```
(defglobal ;/ GLOBAL VARIABLES
    ?*ang-lim* = 0.0
    ?*ngen* = 0
    ?*count* = 0
    ?*max-sen* = 0.0
    ?*plus-cont* = fyhl
    ?*dgen-plus* = 0.0
    ?*ang-bus1* = fyh
    ?*max-ang* = 0.0
    ?*max-dev* = 0.0
    ?*dev-angl* = fyh)


(deftemplate generate ;/ DATA BASE FOR GENERATORS
    (slot gen-num (default ?NONE))
    (slot gen-loc (default ?NONE))
    (slot max-lim (default ?NONE))
    (slot min-lim (default ?NONE))
    (slot pr-set (default ?NONE)))


(deftemplate sensit ;/ DATA BASE FOR SENSITIVITY COEFFICIENTS
    (slot dep-var (default ?NONE))
    (slot bus-name (default ?NONE))
    (slot ctrl-var (default ?NONE))
    (slot ctrl-loc (default ?NONE))
    (slot sen-fact (default ?NONE)))


(deftemplate angle ;/ DATA BASE FOR GENERATOR VOLTAGE ANGLES
    (slot bus-name (default ?NONE))
    (slot bus-type (default ?NONE))
    (slot volt-ang (default ?NONE))
    (slot pr-dev (type NUMBER) (default 0.0))
    (slot priorty (type NUMBER) (default 0))
    (slot seq (type NUMBER) (default 0)))


(defrule open-file-angle ;/ THIS RULE OPEN THE FILE CONTAINING INITIAL
    (initial-fact)       ;/ VALUES OF VOLTAGE ANGLES OBTAINED FROM L.F.
=>
    (open "angle.dat" angl "r")
    (assert (read-file-angle)))


(defrule read-file-angle  ;/ THIS RULE READS THE INITIAL VALUES OF VOLTAGE
```

```
?read-angle <- (read-file-angle) ;/ ANGLES FROM A FILE
=>
     (retract ?read-angle)
     (bind ?bus-name (read angl))
     (bind ?bus-type (read angl))
     (bind ?volt-ang (read angl))
     (if (neq ?bus-name EOF-ANGLE) ; 'if-1'
     then
     (bind ?*ngen* (+ ?*ngen* 1))
     (assert (angle (bus-name ?bus-name)
                    (bus-type ?bus-type)
                    (volt-ang ?volt-ang)))
     (assert (read-file-angle))
     ); close 'if-1'
     (if (eq ?bus-name EOF-ANGLE)
     then
     (assert (EOF-ANGLE))
     (assert (angle-lim))
     (close angl); close the file
     ); close 'if-2'
     ); close the rule


(defrule read-angle-lim  ;/ THIS RULE READS THE DESIRED ANGLE LIMIT
     ?angle <- (angle-lim);/ FROM THE USER
=>
     (retract ?angle)
     (printout t "***** PLEASE ENTER YOUR DESIRED ANGLE LIMIT *****"
      crlf)
     (bind ?*ang-lim* (read))
     (assert (angle-dev))
     ); close the rule


(defrule check-angle-dev ;/ THIS RULE IDENTFIS THE BUSES WITHOUT ANGLE
     ?angle <- (angle-dev);/ DEVIATION
     ?angle-1 <- (angle (bus-name ?bus-name)
                        (bus-type PV)
                        (volt-ang ?volt-ang)
                        (pr-dev ?pr-dev))
     ?angle-2 <- (angle (bus-name ?bus-slk)
                        (bus-type SLK) (volt-ang ?ang-slk))
=>
     (retract ?angle)
     (bind ?*count* (+ ?*count* 1))
     (bind ?pr-dev (abs (- ?ang-slk ?volt-ang)))
     (if (<= ?pr-dev ?*ang-lim*)
     then
     (bind ?pr-dev 0.0))
     (modify ?angle-1 (pr-dev ?pr-dev))
     (if (> ?*ngen* ?*count*)
     then
```

```
    (assert (angle-dev))
    ) ; close 'if'
    ) ; close the rule


(defrule angle-violation-check ;/ THIS RULE CHECKS VOLTAGE ANGLE
    ?initial <- (initial-fact) ;/ VIOLATION
    ?angle-1 <- (angle (bus-name ?bus-name)
                       (bus-type PV)
                       (volt-ang ?volt-ang) (pr-dev ?pr-dev&~0.0))
=>
    (retract ?initial)
    (assert (violation-exist)))


(defrule print-no-violation ;/ THIS RULE PRINTS " NO VIOLATION " MESSAGE
(declare (salience -5)) ;/ IN CASE NO VIOLATION EXIST
    (not (violation-exist))
=>
    (printout t "THERE IS NO VIOLATION IN VOLTAGE ANGLE " crlf))


(defrule open-file-violation ;/ THIS RULE OPEN GENERATION & SENSITIVITY
    (violation-exist)        ;/ DATA FILES IN CASE VIOLATION EXIST
=>
    (open "gen.dat" gen "r")
    (open "angsen.dat" angsen "r")
    (assert (read-file-gen)
            (read-file-angsen)))


(defrule read-file-gen       ;/ READS THE GENERATION DATA FROM A FILE
?read-gen <- (read-file-gen)
=>
    (retract ?read-gen)
    (bind ?gen-num (read gen))
    (bind ?gen-loc (read gen))
    (bind ?max-lim (read gen))
    (bind ?min-lim (read gen))
    (bind ?pr-set (read gen))
    (if (neq ?gen-num EOF-GEN) ; 'if-1'
    then
    (assert (generate (gen-num ?gen-num)
                      (gen-loc ?gen-loc)
                      (max-lim ?max-lim)
                      (min-lim ?min-lim)
                      (pr-set ?pr-set)))
    (assert (read-file-gen))
    ) ; close 'if-1'
    (if (eq ?gen-num EOF-GEN); 'if-2'
    then
```

```
    (assert (EOF-GEN))
    (close gen) ; close the file
    ) ; close 'if-2'
    ) ; close the rule


(defrule read-file-angsen  ;/ READS THE SENSITIVITY DATA FROM A FILE
?read-ang <- (read-file-angsen)
=>
     (retract ?read-ang)
     (bind ?dep-var (read angsen))
     (bind ?bus-name (read angsen))
     (bind ?ctrl-var (read angsen))
     (bind ?ctrl-loc (read angsen))
     (bind ?sen-fact (read angsen))
     (if (neq ?dep-var EOF-ANGSEN) ; 'if-1'
     then
     (assert (sensit (dep-var ?dep-var)
                     (bus-name ?bus-name)
                     (ctrl-var ?ctrl-var)
                     (ctrl-loc ?ctrl-loc)
                     (sen-fact ?sen-fact)))
     (assert (read-file-angsen))
     ) ; close 'if-1'
     (if (eq ?dep-var EOF-ANGSEN); 'if-2'
     then
     (assert (EOF-ANGSEN))
     (close angsen) ; close the file
     ) ; close 'if-2'
     ) ; close the rule


(defrule angle-check-1 ;/ THIS RULE & NEXT ONE IDENTFY THE BUS WITH
     (EOF-ANGLE)        ;/ MAXIMUM VIOLATION
     (EOF-GEN)
     (EOF-ANGSEN)
     ?angle-2 <- (angle (bus-name ?bus-name)
                        (bus-type PV)
                        (volt-ang ?volt-ang)
                        (pr-dev ?pr-dev&~0.0))
     (not (update))
=>
     (assert (max-angle))
     (bind ?*max-dev* (max ?*max-dev* ?pr-dev))
     (if (= ?*max-dev* ?pr-dev)
     then
     (bind ?*max-ang* ?volt-ang)
     (bind ?*ang-bus1* ?bus-name)))


 (defrule angle-check-2
      (declare (salience -5))
```

123-6

```
    ?maxx <- (max-angle)
=>
    (retract ?maxx)
    (bind ?ang-bus ?*ang-bus1*)
    (bind ?volt-ang ?*max-ang*)
    (bind ?*ang-bus1* fyh)
    (bind ?*max-ang* 0.0)
    (bind ?*max-dev* 0.0)
    (if (> ?volt-ang 0.0)
    then
    (assert (ang-bus-plus ?ang-bus ?volt-ang))
    else
    (assert (ang-bus-neg ?ang-bus ?volt-ang))
    ) ; close 'if'
    (assert (gen-max-sen-plus FYH)
            (update)))


(defrule max-gen-plus ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR AND
                      ;/ HAS MARGIN FOR INCREASE IN GENERATION
    ?angle-1 <- (ang-bus-neg ?ang-bus ?)
    ?gen <- (generate (gen-loc ?gen-loc) (max-lim ?max-lim)
                      (pr-set ?pr-set&:(< ?pr-set ?max-lim)))
    ?ang-sens <- (sensit (dep-var ?dep-var)
                         (bus-name ?ang-bus) (ctrl-loc ?gen-loc)
                         (sen-fact ?sen-fact))
=>
    (assert (max-gen-plus))
    (bind ?*max-sen* (max ?*max-sen* ?sen-fact))
    (if (= ?*max-sen* ?sen-fact)
    then
    (bind ?*plus-cont* ?gen-loc)))


(defrule max-gen-minus ;/ IDENTIFIES THE MOST EFFECTIVE GENERATOR AND
                       ;/ HAS MARGIN FOR DECREASE IN GENERATION
    ?angle-1 <- (ang-bus-plus ?ang-bus ?)
    ?gen <- (generate (gen-loc ?gen-loc) (min-lim ?min-lim)
                      (pr-set ?pr-set&:(> ?pr-set ?min-lim)))
    ?ang-sens <- (sensit (dep-var ?dep-var)
                         (bus-name ?ang-bus) (ctrl-loc ?gen-loc)
                         (sen-fact ?sen-fact))
=>
    (assert (max-gen-plus))
    (bind ?*max-sen* (max ?*max-sen* ?sen-fact))
    (if (= ?*max-sen* ?sen-fact)
    then
    (bind ?*plus-cont* ?gen-loc)))
```

```
(defrule nearby-gen ;/ STORES THE MOST EFFECTIVE GENERATOR WITH PLUS
                    ;/ MARGIN
     (declare (salience -10))
     ?max-gen <- (max-gen-plus)
     ?gen-plus <- (gen-max-sen-plus ?)
=>
     (retract ?gen-plus)
     (retract ?max-gen)
     (assert (gen-max-sen-plus ?*plus-cont*))
     (bind ?*max-sen* 0.0)
     (bind ?*plus-cont* fyh1)
     ) ; close the rule


(defrule gen-reschedule-plus ;/ DETERMINES THE AMOUNT OF INCREASE IN
                             ;/ GENERATION FOR THE MOST EFFECTIVE GEN.
     ?angle-1 <- (ang-bus-neg ?ang-bus ?volt-ang)
     ?gen-pluss <- (gen-max-sen-plus ?gen-plus&~FYH1)
     ?angle-2 <- (angle (bus-name ?bus-slk)
                        (bus-type SLK) (volt-ang ?ang-slk))
     ?ang-sens <- (sensit (dep-var ?dep-var)
                          (bus-name ?ang-bus) (ctrl-loc ?gen-plus)
                          (sen-fact ?sen-fact))
     ?gen <- (generate (gen-loc ?gen-plus)
                       (max-lim ?max-lim) (pr-set ?pr-set))
=>
     (retract ?angle-1)
     (retract ?gen-pluss)
     (bind ?d-ang (- ?*ang-lim* (- ?volt-ang ?ang-slk)))
     (bind ?d-gen (/ ?d-ang ?sen-fact))
     (bind ?gen-av (- ?max-lim ?pr-set))
     (if (<= ?d-gen ?gen-av)
     then
     (assert (dgen-plus ?d-gen)
             (gen-plus ?gen-plus) (count 1))
     (bind ?pr-set (+ ?pr-set ?d-gen))
     (modify ?gen (pr-set ?pr-set))
     else
     (assert (dgen-plus ?gen-av)
             (gen-plus ?gen-plus) (count 1))
     (bind ?pr-set (+ ?pr-set ?gen-av))
     (modify ?gen (pr-set ?pr-set))
     ) ; close 'if'
     ) ; close the rule


(defrule gen-reschedule-minus ;/ DETERMINES THE AMOUNT OF DECREASE IN
                              ;/ GENERATION FOR THE MOST EFFECTIVE GEN.
     ?angle-1 <- (ang-bus-plus ?ang-bus ?volt-ang)
     ?gen-pluss <- (gen-max-sen-plus ?gen-plus&~FYH1)
     ?angle-2 <- (angle (bus-name ?bus-slk)
```

```
                    (bus-type SLK) (volt-ang ?ang-slk))
    ?ang-sens <- (sensit (dep-var ?dep-var)
                         (bus-name ?ang-bus) (ctrl-loc ?gen-plus)
                         (sen-fact ?sen-fact))
    ?gen <- (generate (gen-loc ?gen-plus)
                      (min-lim ?min-lim) (pr-set ?pr-set))
=>
    (retract ?angle-1)
    (retract ?gen-pluss)
    (bind ?d-ang (- ?*ang-lim* (- ?volt-ang ?ang-slk)))
    (bind ?d-gen (/ ?d-ang ?sen-fact))
    (bind ?gen-av (- ?min-lim ?pr-set))
    (if (>= ?d-gen ?gen-av)
    then
    (assert (dgen-plus ?d-gen)
            (gen-plus ?gen-plus) (count 1))
    (bind ?pr-set (+ ?pr-set ?d-gen))
    (modify ?gen (pr-set ?pr-set))
    else
    (assert (dgen-plus ?gen-av)
            (gen-plus ?gen-plus) (count 1))
    (bind ?pr-set (+ ?pr-set ?gen-av))
    (modify ?gen (pr-set ?pr-set))
    ) ; close 'if'
    ) ; close the rule


(defrule update-volt-angle ;/ UPDATES THE VOLTAGE ANGLES FOR ALL BUSES
    ?update <- (update)    ;/ BASED ON THE CHANGE IN GENERATION
    ?gen1 <- (gen-plus ?gen)
    ?dgen1 <- (dgen-plus ?fdgen)
    ?angle1 <- (angle (bus-name ?bus-name)
                      (volt-ang ?volt-ang) (pr-dev ?pr-dev))
    ?angle-2 <- (angle (bus-name ?bus-slk)
                       (bus-type SLK) (volt-ang ?ang-slk))
    ?ang-sens <- (sensit (dep-var ?dep-var)
                         (bus-name ?bus-name) (ctrl-loc ?gen)
                         (sen-fact ?sen-fact))
    ?count1 <- (count ?count)
=>
    (bind ?count (+ ?count 1))
    (bind ?d-ang (* ?fdgen ?sen-fact))
    (bind ?volt-ang (+ ?volt-ang ?d-ang))
    (bind ?pr-dev (abs (- ?ang-slk ?volt-ang)))
    (if (<= ?pr-dev ?*ang-lim*)
    then
    (bind ?pr-dev 0.0))
    (modify ?angle1 (volt-ang ?volt-ang) (pr-dev ?pr-dev))
    (if (<= ?count (- ?*ngen* 1))  ; 'if-2'
    then
    (retract ?count1)
    (assert (count ?count))
```

```
    else
    (retract ?update)
    (retract ?gen1)
    (retract ?dgen1)
    (retract ?count1)
    (assert (update-3))
    ) ; close 'if-2'
    ) ; close the rule


(defrule sucess-ful
     (declare (salience -19))
     (update)
     ?update <- (update-3)
=>
     (retract ?update))


(defrule open-print ;/ OPENS THE OUTPUT FILE
     (declare (salience -20))
     (or
     (update-3)
     (update))
=>
     (open "angle.out" angle1 "w"))


(defrule print-title1 ;/ THIS RULE & THE FOLLOWING PRINTS THE RESULTS
(declare (salience -24)) ;/ IN OUTPUT FILE
     (or
     (update-3)
     (update))
=>
     (printout angle1 "EXPERT SYSTEM RESULTS FOR VOLTAGE ANGLE ADJUSTMENT" crlf)
     (printout angle1 " " crlf)
     (printout angle1 "The New Values of the Voltage Angles are as follow:-" crlf
     "  " crlf))


(defrule print-new-angle
     (declare (salience -25))
     (update-3)
     ?angle1 <- (angle (bus-name ?bus-name)
                       (bus-type ?bus-type)
                       (volt-ang ?volt-ang))
=>
     (format angle1 " BUS NAME :%-3d%-10s%-3s%-15s%-4.4f%n"
      ?bus-name "  BUS TYPE : " ?bus-type
     "  VOLTAGE ANGLE : " ?volt-ang ))
```

```
(defrule print-gen-title
     (declare (salience -29))
     (update-3)
=>
     (printout angle1 " " crlf)
     (printout angle1 "The New Settings of Generators are as follow:-" crlf))


(defrule print-new-gener
     (declare (salience -30))
     (update-3)
     ?gen <- (generate (gen-loc ?gen-loc)
                       (max-lim ?max-lim)
                       (min-lim ?min-lim) (pr-set ?pr-set))
=>
   (printout angle1 " " crlf)
   (format angle1 " GEN. LOCAT. :%-3d%-10s%-4.4f%-10s%-4.4f%-10s%-4.4f%n" ?gen-lo
   "  PR. SETT. : " ?pr-set " MAX. GEN. : " ?max-lim " MIN. GEN. : " ?min-lim ))


(defrule print-no-controller ;/ THIS RULE FIRES INCASE NO ADEQUATE CONTROLS
(declare (salience -35))
     (update)
=>
     (printout angle1 " " crlf)
     (printout angle1 " NO ADEQUTE CONTROLLER AVAILABLE. " crlf))


(defrule close-print ;/ CLOSES THE OUTPUT FILE
     (declare (salience -40))
     (or
     (update-3)
     (update))
=>
     (close angle1))
```

123-6

APPENDIX - C

COMPUTER OUTPUTS FOR CASE STUDIES

APPENDIX - C.1

LOAD FLOW PROGRAM OUTPUTS

STUDY CASE NO.1, "14-BUS SYSTEM

###
COMPLETE LOAD FLOW RESULTS (INITIAL STATE)
###

PROGRAM CONVERGED IN 3 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.050	0.000	238.248	-11.649	0.000	0.000
2	PV	1.025	-5.034	40.000	-61.879*	21.700	12.700
3	PV	1.080	-13.935	0.000	103.702*	94.200	19.000
6	PV	1.000	-17.258	0.000	47.542	11.200	7.500
8	PV	1.100	-14.493	0.000	26.255*	0.000	0.000
4	PQ	1.039	-10.938	0.000	0.000	37.800	-3.900
5	PQ	1.035	-9.476	0.000	0.000	7.600	1.600
7	PQ	1.058*	-14.493	0.000	0.000	0.000	0.000
9	PQ	1.057*	-16.283	0.000	19.000	10.500	-6.600
10	PQ	1.040	-16.773	0.000	0.000	9.000	5.800
11	PQ	1.017	-17.172	0.000	0.000	3.500	1.800
12	PQ	0.969	-18.664	0.000	0.000	6.100	1.600

13	PQ	0.946*	-18.826	0.000	0.000	43.500	35.800
14	PQ	0.990	-18.439	0.000	0.000	14.900	5.000

##

GENERATORS/SYNCHRONOUS CONDENSERS

##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	238.248	-11.649			40.000	-40.000
2	PV	40.000	-61.879	100.000	30.000	50.000	-40.000
3	PV	0.000	103.702	0.000	0.000	40.000	0.000
6	PV	0.000	47.542	0.000	0.000	50.000	-6.000
8	PV	0.000	26.255	0.000	0.000	24.000	-6.000

##

STATIC VAR COMPANSATORS

##

LOCATION (BUS)	TYPE	PR. SETT. (MVAR)	MVAR LIMITS (MAX.)	(MIN.)
9	SH. CAPACITOR	19.000	19.000	0.000
13	SH. CAPACITOR	0.000	9.000	0.000
14	SH. REACTOR	0.000	-9.000	0.000

##

TRANSFORMER TAP CHANGERS

##

TAP ID.	BETWEEN BUS	- BUS	TAP LOCATIN	TAP POSITION	LIMITS (MAX.)	(MIN.)
1	4	7	4	0.9750	1.1000	0.9000
2	9	4	9	0.9625	1.1000	0.9000
3	5	6	5	0.9375	1.1000	0.9000

STUDY CASE NO.1, "14-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (FINAL STATE)
##

PROGRAM CONVERGED IN 3 ITERATIONS

BUS	TYPE	BUS VOLT.(KV) MAGN.	ANGLE	GENERATION MW	MVAR	LOAD MW	MVAR
1	SLK	1.086	0.000	234.743	8.848	0.000	0.000
2	PV	1.057	-4.575	40.000	3.834	21.700	12.700
3	PV	1.025	-12.105	0.000	22.051	94.200	19.000
6	PV	1.000	-16.498	0.000	45.132	11.200	7.500
8	PV	1.081	-13.325	0.000	24.676*	0.000	0.000
4	PQ	1.038	-9.847	0.000	0.000	37.800	-3.900
5	PQ	1.044	-8.584	0.000	0.000	7.600	1.600
7	PQ	1.041	-13.325	0.000	0.000	0.000	0.000
9	PQ	1.032	-15.117	0.000	0.000	10.500	-6.600
10	PQ	1.019	-15.678	0.000	0.000	9.000	5.800
11	PQ	1.007	-16.232	0.000	0.000	3.500	1.800
12	PQ	0.971	-17.962	0.000	0.000	6.100	1.600

13	PQ	0.950	-18.236	0.000	9.000	43.500	35.800
14	PQ	0.977	-17.551	0.000	0.000	14.900	5.000

##

GENERATORS/SYNCHRONOUS CONDENSERS

##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	234.743	8.848			40.000	-40.000
2	PV	40.000	3.834	100.000	30.000	50.000	-40.000
3	PV	0.000	22.051	0.000	0.000	40.000	0.000
6	PV	0.000	45.132	0.000	0.000	50.000	-6.000
8	PV	0.000	24.676	0.000	0.000	24.000	-6.000

##

STATIC VAR COMPANSATORS

##

LOCATION (BUS)	TYPE	PR. SETT. (MVAR)	MVAR LIMITS (MAX.)	(MIN.)
9	SH. CAPACITOR	0.000	19.000	0.000
13	SH. CAPACITOR	9.000	9.000	0.000
14	SH. REACTOR	0.000	-9.000	0.000

##

TRANSFORMER TAP CHANGERS

##

TAP ID.	BETWEEN BUS	- BUS	TAP LOCATIN	TAP POSITION	LIMITS (MAX.)	(MIN.)
1	4	7	4	0.9750	1.1000	0.9000
2	9	4	9	0.9625	1.1000	0.9000
3	5	6	5	0.9375	1.1000	0.9000

STUDY CASE NO.2, "30-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (INITIAL STATE)
##

PROGRAM CONVERGED IN 3 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.060	0.000	347.071	-32.779	0.000	0.000
2	PV	1.045	-7.323	40.000	76.664*	21.700	12.700
5	PV	1.010	-17.328	0.000	39.084	94.200	19.000
8	PV	1.010	-16.300	0.000	54.567*	30.000	30.000
11	PV	1.082	-21.815	0.000	41.919*	0.000	0.000
13	PV	1.071	-25.224	0.000	69.618*	0.000	0.000
3	PQ	1.018	-11.018	0.000	0.000	2.400	1.200
4	PQ	1.010	-13.342	0.000	0.000	7.600	1.600
6	PQ	1.004	-15.407	0.000	0.000	0.000	0.000
7	PQ	0.999	-16.727	0.000	0.000	22.800	10.900
9	PQ	1.001	-21.815	0.000	0.000	0.000	0.000
10	PQ	0.975	-25.212	0.000	19.000	5.800	2.000

12	PQ	0.980	-25.224	0.000	0.000	11.200	7.500
14	PQ	0.948*	-27.190	0.000	0.000	6.200	1.600
15	PQ	0.921*	-27.857	0.000	0.000	28.200	12.500
16	PQ	0.971	-25.544	0.000	0.000	3.500	1.800
17	PQ	0.968	-25.553	0.000	0.000	9.000	5.800
18	PQ	0.895*	-29.157	0.000	0.000	3.200	0.900
19	PQ	0.884*	-29.699	0.000	0.000	29.500	13.400
20	PQ	0.900*	-28.864	0.000	0.000	12.200	6.700
21	PQ	0.958	-25.861	0.000	0.000	17.500	11.200
22	PQ	0.957	-25.887	0.000	0.000	0.000	0.000
23	PQ	0.900*	-29.026	0.000	0.000	23.200	5.600
24	PQ	0.924*	-26.783	0.000	4.300	8.700	6.700
25	PQ	0.930*	-24.469	0.000	0.000	0.000	0.000
26	PQ	0.911*	-24.973	0.000	0.000	3.500	2.300
27	PQ	0.944*	-22.739	0.000	0.000	0.000	0.000
28	PQ	1.001	-16.264	0.000	0.000	0.000	0.000
29	PQ	0.922*	-24.190	0.000	0.000	2.400	0.900
30	PQ	0.910*	-25.238	0.000	0.000	10.600	1.900

##
GENERATORS/SYNCHRONOUS CONDENSERS
##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	PRES. GEN. MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	347.071	-32.779			40.000	-40.000
2	PV	40.000	76.664	100.000	30.000	50.000	-40.000
5	PV	0.000	39.084	0.000	0.000	40.000	-40.000
8	PV	0.000	54.567	0.000	0.000	50.000	-10.000
11	PV	0.000	41.919	0.000	0.000	35.000	-6.000
13	PV	0.000	69.618	0.000	0.000	50.000	-6.000

##
STATIC VAR COMPANSATORS
##

LOCATION (BUS)	TYPE	PR. SETT. (MVAR)	MVAR LIMITS (MAX.)	(MIN.)
10	SH. CAPACITOR	19.000	19.000	0.000
15	SH. CAPACITOR	0.000	25.000	0.000
24	SH. CAPACITOR	4.300	8.600	0.000
28	SH. REACTOR	0.000	-6.000	0.000
16	SH. REACTOR	0.000	-10.000	0.000

##
TRANSFORMER TAP CHANGERS
##

TAP ID.	BETWEEN BUS	- BUS	TAP LOCATIN	TAP POSITION	LIMITS (MAX.)	(MIN.)
1	4	12	4	0.9250	1.1000	0.9000
2	6	9	6	0.9750	1.1000	0.9000
3	10	6	10	0.9625	1.1000	0.9000
4	28	27	28	0.9625	1.1000	0.9000

STUDY CASE NO.2, "30-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (FINAL STATE)
##

PROGRAM CONVERGED IN 3 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.094	0.000	343.965	3.034	0.000	0.000
2	PV	1.062	-6.590	40.000	32.722	21.700	12.700
5	PV	1.031	-16.225	0.000	39.638	94.200	19.000
8	PV	1.029	-15.168	0.000	44.513	30.000	30.000
11	PV	1.100	-20.202	0.000	28.075	0.000	0.000
13	PV	1.095	-23.273	0.000	51.225*	0.000	0.000
3	PQ	1.045	-10.276	0.000	0.000	2.400	1.200
4	PQ	1.035	-12.447	0.000	0.000	7.600	1.600
6	PQ	1.026	-14.367	0.000	0.000	0.000	0.000
7	PQ	1.020	-15.640	0.000	0.000	22.800	10.900
9	PQ	1.047	-20.202	0.000	0.000	0.000	0.000
10	PQ	1.027	-23.319	0.000	19.000	5.800	2.000

123-6

12	PQ	1.030	-23.273	0.000	0.000	11.200	7.500
14	PQ	1.010	-25.215	0.000	0.000	6.200	1.600
15	PQ	0.995	-26.233	0.000	25.000	28.200	12.500
16	PQ	1.022	-23.587	0.000	0.000	3.500	1.800
17	PQ	1.020	-23.617	0.000	0.000	9.000	5.800
18	PQ	0.965	-27.160	0.000	0.000	3.200	0.900
19	PQ	0.950	-27.509	0.000	0.000	29.500	13.400
20	PQ	0.963	-26.720	0.000	0.000	12.200	6.700
21	PQ	1.012	-23.954	0.000	0.000	17.500	11.200
22	PQ	1.012	-23.994	0.000	0.000	0.000	0.000
23	PQ	0.971	-27.140	0.000	0.000	23.200	5.600
24	PQ	0.986	-25.064	0.000	8.600	8.700	6.700
25	PQ	0.978	-22.762	0.000	0.000	0.000	0.000
26	PQ	0.960	-23.216	0.000	0.000	3.500	2.300
27	PQ	0.983	-21.051	0.000	0.000	0.000	0.000
28	PQ	1.024	-15.184	0.000	0.000	0.000	0.000
29	PQ	0.962	-22.387	0.000	0.000	2.400	0.900
30	PQ	0.950*	-23.348	0.000	0.000	10.600	1.900

GENERATORS/SYNCHRONOUS CONDENSERS

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	PRES. GEN. MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	343.965	3.034			40.000	-40.000
2	PV	40.000	32.722	100.000	30.000	50.000	-40.000
5	PV	0.000	39.638	0.000	0.000	40.000	-40.000
8	PV	0.000	44.513	0.000	0.000	50.000	-10.000
11	PV	0.000	28.075	0.000	0.000	35.000	-6.000
13	PV	0.000	51.225	0.000	0.000	50.000	-6.000

STATIC VAR COMPANSATORS

LOCATION (BUS)	TYPE	PR. SETT. (MVAR)	MVAR LIMITS (MAX.)	(MIN.)
10	SH. CAPACITOR	19.000	19.000	0.000
15	SH. CAPACITOR	25.000	25.000	0.000
24	SH. CAPACITOR	8.600	8.600	0.000
28	SH. REACTOR	0.000	-6.000	0.000
16	SH. REACTOR	0.000	-10.000	0.000

TRANSFORMER TAP CHANGERS

TAP ID.	BETWEEN BUS	- BUS	TAP LOCATIN	TAP POSITION	LIMITS (MAX.)	(MIN.)
1	4	12	4	0.9375	1.1000	0.9000
2	6	9	6	1.0125	1.1000	0.9000
3	10	6	10	0.9625	1.1000	0.9000
4	28	27	28	0.9625	1.1000	0.9000

STUDY CASE NO.1, "6-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (INITIAL STATE)
##

PROGRAM CONVERGED IN 2 ITERATIONS

BUS	TYPE	BUS VOLT.(KV) MAGN.	ANGLE	GENERATION MW	MVAR	LOAD MW	MVAR
1	SLK	241.500	0.000	107.869	15.955*	0.000	0.000
2	PV	241.500	-3.671	50.000	74.350*	0.000	0.000
3	PV	246.100	-4.273	60.000	89.624*	0.000	0.000
4	PQ	227.556	-4.196	0.000	0.000	70.000	70.000
5	PQ	226.653	-5.276	0.000	0.000	70.000	70.000
6	PQ	231.018	-5.947	0.000	0.000	70.000	70.000

##

LINE		LINE FLOW MW	MVAR	LINE LOSSES MW	MVAR	FLOW LIM. MW
1	-TO- 2	28.687	-15.418	0.905	-2.600	33.500
1	-TO- 4	43.583*	20.119	1.087	0.187	36.000
1	-TO- 5	35.599*	11.254	1.073	-2.196	30.000
2	-TO- 1	-27.782	12.817	0.905	-2.600	33.500
2	-TO- 3	2.930	-12.269	0.040	-6.541	15.500
2	-TO- 4	33.091	46.051	1.505	0.929	60.000
2	-TO- 5	15.514	15.352	0.498	-2.654	30.000
2	-TO- 6	26.248	12.399	0.583	-3.612	40.000
3	-TO- 2	-2.890	5.728	0.040	-6.541	15.500

123-6

3	-TO- 5	19.117	23.173	1.093	-2.921	23.000
3	-TO- 6	43.773*	60.723	1.003	2.863	40.000
4	-TO- 1	-42.495*	-19.932	1.087	0.187	36.000
4	-TO- 2	-31.586	-45.123	1.505	0.929	60.000
4	-TO- 5	4.083	-4.942	0.036	-7.727	30.000
5	-TO- 1	-34.525	-13.449	1.073	-2.196	30.000
5	-TO- 2	-15.016	-18.005	0.498	-2.654	30.000
5	-TO- 3	-18.023	-26.094	1.093	-2.921	23.000
5	-TO- 4	-4.047	-2.785	0.036	-7.727	30.000
5	-TO- 6	1.614	-9.663	0.050	-5.791	15.000
6	-TO- 2	-25.665	-16.011	0.583	-3.612	40.000
6	-TO- 3	-42.770*	-57.860	1.003	2.863	40.000
6	-TO- 5	-1.565	3.871	0.050	-5.791	15.000

##

##

GENERATORS/SYNCHRONOUS CONDENSERS

##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	107.869	15.955			0.000	0.000
2	PV	50.000	74.350	150.000	37.500	0.000	0.000
3	PV	60.000	89.624	180.000	45.000	0.000	0.000

123-6

STUDY CASE NO.1, "6-BUS SYSTEM"

```
############################################################################
                COMPLETE LOAD FLOW RESULTS (FINAL STATE)
############################################################################
```

PROGRAM CONVERGED IN 2 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	241.500	0.000	58.186	32.821	0.000	0.000
2	PV	241.500	-0.712	108.308	52.099	0.000	0.000
3	PV	246.100	-2.256	50.486	92.132	0.000	0.000
4	PQ	227.061	-2.294	0.000	0.000	70.000	70.000
5	PQ	226.496	-3.469	0.000	0.000	70.000	70.000
6	PQ	230.886	-3.717	0.000	0.000	70.000	70.000

```
############################################################################
```

		LINE FLOW		LINE LOSSES		FLOW LIM.
LINE		MW	MVAR	MW	MVAR	MW
1	-TO- 2	5.495	-4.910	0.034	-4.342	33.500
1	-TO- 4	27.378	24.324	0.659	-1.518	36.000
1	-TO- 5	25.313	13.406	0.668	-3.713	30.000
2	-TO- 1	-5.461	0.568	0.034	-4.342	33.500
2	-TO- 3	10.062	-13.557	0.094	-6.274	15.500
2	-TO- 4	49.423	40.498	1.893	1.708	60.000
2	-TO- 5	21.892	13.729	0.665	-2.150	30.000
2	-TO- 6	32.392	10.860	0.784	-3.036	40.000
3	-TO- 2	-9.969	7.282	0.094	-6.274	15.500

123-6

3	-TO- 5	20.453	22.866	1.132	-2.833	23.000
3	-TO- 6	40.002	61.984	0.976	2.726	40.000
4	-TO- 1	-26.719	-25.842	0.659	-1.518	36.000
4	-TO- 2	-47.530	-38.790	1.893	1.708	60.000
4	-TO- 5	4.250	-5.366	0.041	-7.694	30.000
5	-TO- 1	-24.645	-17.119	0.668	-3.713	30.000
5	-TO- 2	-21.227	-15.879	0.665	-2.150	30.000
5	-TO- 3	-19.321	-25.699	1.132	-2.833	23.000
5	-TO- 4	-4.209	-2.328	0.041	-7.694	30.000
5	-TO- 6	-0.597	-8.972	0.038	-5.818	15.000
6	-TO- 2	-31.609	-13.895	0.784	-3.036	40.000
6	-TO- 3	-39.026	-59.258	0.976	2.726	40.000
6	-TO- 5	0.635	3.154	0.038	-5.818	15.000

##

##
GENERATORS/SYNCHRONOUS CONDENSERS
##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	58.186	32.821			0.000	0.000
2	PV	108.308	52.099	150.000	37.500	0.000	0.000
3	PV	50.486	92.132	180.000	45.000	0.000	0.000

123-6

STUDY CASE NO.2, "25-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (INITIAL STATE)
##

PROGRAM CONVERGED IN 2 ITERATIONS

		BUS VOLT. (KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.020	0.000	245.687	125.868*	200.000	65.000
2	PV	1.020	9.461	100.000	10.455	10.000	3.000
3	PV	1.010	6.851	150.000	15.807	50.000	17.000
4	PV	1.010	-1.271	50.000	46.377*	30.000	10.000
5	PV	1.015	9.242	200.000	-8.616	25.000	8.000
6	PQ	0.991	5.881	0.000	0.000	15.000	5.000
7	PQ	1.001	4.882	0.000	0.000	15.000	5.000
8	PQ	1.000	4.331	0.000	0.000	25.000	0.000
9	PQ	0.990	3.715	0.000	0.000	15.000	5.000
10	PQ	1.000	5.015	0.000	0.000	15.000	5.000
11	PQ	1.000	3.825	0.000	0.000	5.000	0.000
12	PQ	1.000	3.616	0.000	0.000	10.000	0.000

13	PQ	0.989	5.587	0.000	0.000	25.000	8.000
14	PQ	0.956	-1.819	0.000	0.000	20.000	7.000
15	PQ	0.956	-2.920	0.000	0.000	30.000	10.000
16	PQ	0.971	-2.777	0.000	0.000	30.000	10.000
17	PQ	1.001	3.305	0.000	0.000	60.000	20.000
18	PQ	0.992	1.002	0.000	0.000	15.000	5.000
19	PQ	1.005	0.309	0.000	0.000	15.000	5.000
20	PQ	0.988	-3.365	0.000	0.000	25.000	8.000
21	PQ	0.977	-4.625	0.000	0.000	20.000	7.000
22	PQ	0.967	-5.706	0.000	0.000	20.000	7.000
23	PQ	0.988	-3.273	0.000	0.000	15.000	5.000
24	PQ	0.963	-7.055	0.000	0.000	15.000	5.000
25	PQ	0.970	-6.488	0.000	0.000	25.000	8.000

##

LINE		LINE FLOW MW	LINE FLOW MVAR	LINE LOSSES MW	LINE LOSSES MVAR	FLOW LIM. MW
1	-TO- 3	-38.774	14.880	1.213	3.003	100.000
1	-TO- 16	40.802	26.795	0.691	-0.055	120.000
1	-TO- 17	-16.161	12.689	0.430	-0.321	60.000
1	-TO- 19	0.115	2.852	0.023	-2.235	50.000

123-6

1	-TO- 23	26.699	-0.784	0.748	-4.231	40.000
1	-TO- 25	33.006	4.437	0.847	-4.610	90.000
2	-TO- 6	22.740	4.857	0.327	-0.327	100.000
2	-TO- 7	33.859*	1.580	0.566	1.122	30.000
2	-TO- 8	33.404	1.018	0.623	1.188	50.000
3	-TO- 1	39.987	-11.878	1.213	3.003	100.000
3	-TO- 13	17.807	7.151	0.207	-0.307	40.000
3	-TO- 14	42.204*	3.534	2.089	4.521	40.000
4	-TO- 19	-43.671*	27.633	0.519	0.215	40.000
4	-TO- 20	39.305	7.103	0.604	-0.603	40.000
4	-TO- 21	24.373	1.640	0.584	-3.976	40.000
5	-TO- 10	31.752	-1.912	0.487	-3.532	55.000
5	-TO- 17	83.496*	-0.674	0.980	-4.927	75.000
5	-TO- 19	59.768	-14.030	3.381	7.460	85.000
6	-TO- 2	-22.413	-5.184	0.327	-0.327	100.000
6	-TO- 13	7.415	0.187	0.015	-0.354	25.000
7	-TO- 2	-33.293*	-0.457	0.566	1.122	30.000
7	-TO- 8	5.972	-2.038	0.020	-0.724	15.000
7	-TO- 12	12.324	-2.498	0.057	-0.830	15.000
8	-TO- 2	-32.781	0.170	0.623	1.188	50.000
8	-TO- 7	-5.951	1.313	0.020	-0.724	15.000
8	-TO- 9	6.564*	3.217	0.022	-1.062	6.000
8	-TO- 17	7.171	-4.693	0.027	-5.593	10.000
9	-TO- 8	-6.542*	-4.279	0.022	-1.062	6.000
9	-TO- 10	-8.457	-0.720	0.071	-0.645	10.000
10	-TO- 5	-31.265	-1.620	0.487	-3.532	55.000
10	-TO- 9	8.528	0.075	0.071	-0.645	10.000
10	-TO- 11	7.741	-3.449	0.061	-1.189	15.000
11	-TO- 10	-7.680	2.260	0.061	-1.189	15.000
11	-TO- 17	2.680	-2.259	0.010	-1.584	20.000
12	-TO- 7	-12.268	1.668	0.057	-0.830	15.000
12	-TO- 17	2.268	-1.667	0.003	-1.377	12.000
13	-TO- 3	-17.600	-7.458	0.207	-0.307	40.000
13	-TO- 6	-7.400	-0.540	0.015	-0.354	25.000
14	-TO- 3	-40.115	0.987	2.089	4.521	40.000
14	-TO- 15	20.124	-7.958	0.143	-0.013	25.000
15	-TO- 14	-19.981	7.945	0.143	-0.013	25.000
15	-TO- 16	-9.994	-17.946	0.111	-1.081	15.000
16	-TO- 1	-40.111	-26.850	0.691	-0.055	120.000
16	-TO- 15	10.105	16.865	0.111	-1.081	15.000
17	-TO- 1	16.591	-13.010	0.430	-0.321	60.000
17	-TO- 5	-82.516*	-4.253	0.980	-4.927	75.000
17	-TO- 8	-7.143	-0.900	0.027	-5.593	10.000
17	-TO- 11	-2.670	0.675	0.010	-1.584	20.000
17	-TO- 12	-2.265	0.290	0.003	-1.377	12.000
17	-TO- 18	18.008*	-2.753	0.265	-0.515	15.000
18	-TO- 17	-17.743*	2.238	0.265	-0.515	15.000
18	-TO- 19	2.745	-7.237	0.045	-1.196	9.000
19	-TO- 1	-0.092	-5.086	0.023	-2.235	50.000
19	-TO- 4	44.190*	-27.418	0.519	0.215	40.000
19	-TO- 5	-56.387	21.490	3.381	7.460	85.000
19	-TO- 18	-2.700	6.041	0.045	-1.196	9.000

20	-TO- 4	-38.701	-7.707	0.604	-0.603	40.000
20	-TO- 21	13.703*	-0.292	0.120	-3.102	12.000
21	-TO- 4	-23.789	-5.616	0.584	-3.976	40.000
21	-TO- 20	-13.584*	-2.810	0.120	-3.102	12.000
21	-TO- 22	17.378	1.428	0.134	-1.897	25.000
22	-TO- 21	-17.244	-3.325	0.134	-1.897	25.000
22	-TO- 23	-10.671	0.366	0.277	-1.177	15.000
22	-TO- 24	7.924	-4.036	0.067	-5.100	7.000
23	-TO- 1	-25.950	-3.447	0.748	-4.231	40.000
23	-TO- 22	10.948	-1.542	0.277	-1.177	15.000
24	-TO- 22	-7.857	-1.063	0.067	-5.100	7.000
24	-TO- 25	-7.131	-3.937	0.029	-2.872	20.000
25	-TO- 1	-32.159	-9.046	0.847	-4.610	90.000
25	-TO- 24	7.160	1.065	0.029	-2.872	20.000

##

##
GENERATORS/SYNCHRONOUS CONDENSERS
##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	245.687	125.868			80.000	-70.000
2	PV	100.000	10.455	150.000	70.000	50.000	-40.000
3	PV	150.000	15.807	200.000	50.000	40.000	-40.000
4	PV	50.000	46.377	60.000	30.000	40.000	-10.000
5	PV	200.000	-8.616	250.000	100.000	70.000	-70.000

STUDY CASE NO.2, "25-BUS SYSTEM"

##
COMPLETE LOAD FLOW RESULTS (FINAL STATE)
##

PROGRAM CONVERGED IN 2 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.020	0.000	297.787	107.609*	200.000	65.000
2	PV	1.020	3.670	74.658	15.037	10.000	3.000
3	PV	1.010	5.264	150.000	14.215	50.000	17.000
4	PV	1.010	-3.757	50.000	44.486*	30.000	10.000
5	PV	1.015	4.348	170.028	-3.956	25.000	8.000
6	PQ	0.989	2.432	0.000	0.000	15.000	5.000
7	PQ	1.001	-0.073	0.000	0.000	15.000	5.000
8	PQ	1.000	-0.503	0.000	0.000	25.000	0.000
9	PQ	0.990	-1.051	0.000	0.000	15.000	5.000
10	PQ	0.999	0.357	0.000	0.000	15.000	5.000
11	PQ	0.999	-0.487	0.000	0.000	5.000	0.000
12	PQ	0.999	-0.873	0.000	0.000	10.000	0.000

13	PQ	0.987	2.738	0.000	0.000	25.000	8.000
14	PQ	0.957	-2.509	0.000	0.000	20.000	7.000
15	PQ	0.957	-3.406	0.000	0.000	30.000	10.000
16	PQ	0.972	-3.090	0.000	0.000	30.000	10.000
17	PQ	1.000	-0.594	0.000	0.000	60.000	20.000
18	PQ	0.992	-2.329	0.000	0.000	15.000	5.000
19	PQ	1.005	-2.407	0.000	0.000	15.000	5.000
20	PQ	0.988	-5.647	0.000	0.000	25.000	8.000
21	PQ	0.977	-6.570	0.000	0.000	20.000	7.000
22	PQ	0.966	-7.179	0.000	0.000	20.000	7.000
23	PQ	0.988	-3.803	0.000	0.000	15.000	5.000
24	PQ	0.963	-8.003	0.000	0.000	15.000	5.000
25	PQ	0.970	-7.147	0.000	0.000	25.000	8.000

##

LINE		LINE FLOW		LINE LOSSES		FLOW LIM.
		MW	MVAR	MW	MVAR	MW
1	-TO- 3	-29.732	11.569	0.720	1.031	100.000
1	-TO- 16	44.483	25.526	0.759	0.264	120.000
1	-TO- 17	5.724	4.617	0.060	-1.343	60.000
1	-TO- 19	11.013	-1.260	0.173	-1.843	50.000

123-6

1	-TO-	23	30.201	-2.094	0.952	-3.808	40.000
1	-TO-	25	36.097	4.251	0.999	-3.881	90.000
2	-TO-	6	- 9.306	7.990	0.099	-1.406	100.000
2	-TO-	7	27.943	2.321	0.388	0.273	30.000
2	-TO-	8	27.412	1.726	0.422	0.229	50.000
3	-TO-	1	30.452	-10.538	0.720	1.031	100.000
3	-TO-	13	31.361	3.854	0.554	0.604	40.000
3	-TO-	14	38.186	3.899	1.718	3.399	40.000
4	-TO-	19	-37.300	23.642	0.380	-0.151	40.000
4	-TO-	20	36.170	8.110	0.522	-0.820	40.000
4	-TO-	21	21.134	2.734	0.454	-4.315	40.000
5	-TO-	10	30.069	-1.506	0.437	-3.766	55.000
5	-TO-	17	69.717	0.508	0.687	-7.492	75.000
5	-TO-	19	45.249	-10.958	1.941	3.673	85.000
6	-TO-	2	-9.207	-9.396	0.099	-1.406	100.000
6	-TO-	13	-5.792	4.397	0.015	-0.352	25.000
7	-TO-	2	-27.554	-2.048	0.388	0.273	30.000
7	-TO-	8	4.684	-1.585	0.012	-0.746	15.000
7	-TO-	12	7.872	-1.365	0.023	-0.991	15.000
8	-TO-	2	-26.990	-1.497	0.422	0.229	50.000
8	-TO-	7	-4.672	0.838	0.012	-0.746	15.000
8	-TO-	9	5.988	3.513	0.020	-1.071	6.000
8	-TO-	17	0.675	-2.850	0.000	-5.716	10.000
9	-TO-	8	-5.967	-4.584	0.020	-1.071	6.000
9	-TO-	10	-9.032	-0.416	0.081	-0.617	10.000
10	-TO-	5	-29.632	-2.260	0.437	-3.766	55.000
10	-TO-	9	9.113	-0.201	0.081	-0.617	10.000
10	-TO-	11	5.521	-2.534	0.031	-1.267	15.000
11	-TO-	10	-5.490	1.267	0.031	-1.267	15.000
11	-TO-	17	0.491	-1.267	0.000	-1.606	20.000
12	-TO-	7	-7.849	0.374	0.023	-0.991	15.000
12	-TO-	17	-2.151	-0.373	0.002	-1.378	12.000
13	-TO-	3	-30.807	-3.250	0.554	0.604	40.000
13	-TO-	6	5.807	-4.748	0.015	-0.352	25.000
14	-TO-	3	-36.468	-0.500	1.718	3.399	40.000
14	-TO-	15	16.474	-6.479	0.095	-0.144	25.000
15	-TO-	14	-16.379	6.336	0.095	-0.144	25.000
15	-TO-	16	-13.599	-16.336	0.120	-1.061	15.000
16	-TO-	1	-43.724	-25.262	0.759	0.264	120.000
16	-TO-	15	13.719	15.275	0.120	-1.061	15.000
17	-TO-	1	-5.664	-5.960	0.060	-1.343	60.000
17	-TO-	5	-69.030	-8.000	0.687	-7.492	75.000
17	-TO-	8	-0.674	-2.866	0.000	-5.716	10.000
17	-TO-	11	-0.490	-0.339	0.000	-1.606	20.000
17	-TO-	12	2.153	-1.005	0.002	-1.378	12.000
17	-TO-	18	13.706	-1.809	0.153	-0.808	15.000
18	-TO-	17	-13.554	1.001	0.153	-0.808	15.000
18	-TO-	19	-1.445	-6.001	0.027	-1.244	9.000
19	-TO-	1	-10.839	-0.583	0.173	-1.843	50.000
19	-TO-	4	37.680	-23.793	0.380	-0.151	40.000
19	-TO-	5	-43.308	14.632	1.941	3.673	85.000
19	-TO-	18	1.473	4.757	0.027	-1.244	9.000

20	-TO- 4	-35.648	-8.930	0.522	-0.820	40.000
20	-TO- 21	10.649	0.931	0.076	-3.216	12.000
21	-TO- 4	-20.680	-7.049	0.454	-4.315	40.000
21	-TO- 20	-10.573	-4.147	0.076	-3.216	12.000
21	-TO- 22	11.257	4.197	0.067	-2.070	25.000
22	-TO- 21	-11.189	-6.267	0.067	-2.070	25.000
22	-TO- 23	-13.766	2.423	0.482	-0.852	15.000
22	-TO- 24	4.965	-3.154	0.026	-5.204	7.000
23	-TO- 1	-29.249	-1.714	0.952	-3.808	40.000
23	-TO- 22	14.247	-3.275	0.482	-0.852	15.000
24	-TO- 22	-4.939	-2.050	0.026	-5.204	7.000
24	-TO- 25	-10.048	-2.951	0.053	-2.798	20.000
25	-TO- 1	-35.098	-8.132	0.999	-3.881	90.000
25	-TO- 24	10.100	0.153	0.053	-2.798	20.000

##

##

GENERATORS/SYNCHRONOUS CONDENSERS

##

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	297.787	107.609			80.000	-70.000
2	PV	74.658	15.037	150.000	70.000	50.000	-40.000
3	PV	150.000	14.215	200.000	50.000	40.000	-40.000
4	PV	50.000	44.486	60.000	30.000	40.000	-10.000
5	PV	170.028	-3.956	250.000	100.000	70.000	-70.000

123-6

STUDY CASE NO.1, "25-BUS SYSTEM" (VOLTAGE ANGLE)

##
COMPLETE LOAD FLOW RESULTS (INITIAL STATE)
##

PROGRAM CONVERGED IN 2 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.020	0.000	245.687	125.868*	200.000	65.000
2	PV	1.020	9.461	100.000	10.455	10.000	3.000
3	PV	1.010	6.851	150.000	15.807	50.000	17.000
4	PV	1.010	-1.271	50.000	46.377*	30.000	10.000
5	PV	1.015	9.242	200.000	-8.616	25.000	8.000
6	PQ	0.991	5.881	0.000	0.000	15.000	5.000
7	PQ	1.001	4.882	0.000	0.000	15.000	5.000
8	PQ	1.000	4.331	0.000	0.000	25.000	0.000
9	PQ	0.990	3.715	0.000	0.000	15.000	5.000
10	PQ	1.000	5.015	0.000	0.000	15.000	5.000
11	PQ	1.000	3.825	0.000	0.000	5.000	0.000
12	PQ	1.000	3.616	0.000	0.000	10.000	0.000

13	PQ	0.989	5.587	0.000	0.000	25.000	8.000
14	PQ	0.956	-1.819	0.000	0.000	20.000	7.000
15	PQ	0.956	-2.920	0.000	0.000	30.000	10.000
16	PQ	0.971	-2.777	0.000	0.000	30.000	10.000
17	PQ	1.001	3.305	0.000	0.000	60.000	20.000
18	PQ	0.992	1.002	0.000	0.000	15.000	5.000
19	PQ	1.005	0.309	0.000	0.000	15.000	5.000
20	PQ	0.988	-3.365	0.000	0.000	25.000	8.000
21	PQ	0.977	-4.625	0.000	0.000	20.000	7.000
22	PQ	0.967	-5.706	0.000	0.000	20.000	7.000
23	PQ	0.988	-3.273	0.000	0.000	15.000	5.000
24	PQ	0.963	-7.055	0.000	0.000	15.000	5.000
25	PQ	0.970	-6.488	0.000	0.000	25.000	8.000

##

LINE		LINE FLOW		LINE LOSSES		FLOW LIM.
		MW	MVAR	MW	MVAR	MW
1	-TO- 3	-38.774	14.880	1.213	3.003	100.000
1	-TO- 16	40.802	26.795	0.691	-0.055	120.000
1	-TO- 17	-16.161	12.689	0.430	-0.321	60.000
1	-TO- 19	0.115	2.852	0.023	-2.235	50.000

123-6

1	-TO- 23	26.699	-0.784	0.748	-4.231	40.000
1	-TO- 25	33.006	4.437	0.847	-4.610	90.000
2	-TO- 6	22.740	4.857	0.327	-0.327	100.000
2	-TO- 7	33.859*	1.580	0.566	1.122	30.000
2	-TO- 8	33.404	1.018	0.623	1.188	50.000
3	-TO- 1	39.987	-11.878	1.213	3.003	100.000
3	-TO- 13	17.807	7.151	0.207	-0.307	40.000
3	-TO- 14	42.204*	3.534	2.089	4.521	40.000
4	-TO- 19	-43.671*	27.633	0.519	0.215	40.000
4	-TO- 20	39.305	7.103	0.604	-0.603	40.000
4	-TO- 21	24.373	1.640	0.584	-3.976	40.000
5	-TO- 10	31.752	-1.912	0.487	-3.532	55.000
5	-TO- 17	83.496*	-0.674	0.980	-4.927	75.000
5	-TO- 19	59.768	-14.030	3.381	7.460	85.000
6	-TO- 2	-22.413	-5.184	0.327	-0.327	100.000
6	-TO- 13	7.415	0.187	0.015	-0.354	25.000
7	-TO- 2	-33.293*	-0.457	0.566	1.122	30.000
7	-TO- 8	5.972	-2.038	0.020	-0.724	15.000
7	-TO- 12	12.324	-2.498	0.057	-0.830	15.000
8	-TO- 2	-32.781	0.170	0.623	1.188	50.000
8	-TO- 7	-5.951	1.313	0.020	-0.724	15.000
8	-TO- 9	6.564*	3.217	0.022	-1.062	6.000
8	-TO- 17	7.171	-4.693	0.027	-5.593	10.000
9	-TO- 8	-6.542*	-4.279	0.022	-1.062	6.000
9	-TO- 10	-8.457	-0.720	0.071	-0.645	10.000
10	-TO- 5	-31.265	-1.620	0.487	-3.532	55.000
10	-TO- 9	8.528	0.075	0.071	-0.645	10.000
10	-TO- 11	7.741	-3.449	0.061	-1.189	15.000
11	-TO- 10	-7.680	2.260	0.061	-1.189	15.000
11	-TO- 17	2.680	-2.259	0.010	-1.584	20.000
12	-TO- 7	-12.268	1.668	0.057	-0.830	15.000
12	-TO- 17	2.268	-1.667	0.003	-1.377	12.000
13	-TO- 3	-17.600	-7.458	0.207	-0.307	40.000
13	-TO- 6	-7.400	-0.540	0.015	-0.354	25.000
14	-TO- 3	-40.115	0.987	2.089	4.521	40.000
14	-TO- 15	20.124	-7.958	0.143	-0.013	25.000
15	-TO- 14	-19.981	7.945	0.143	-0.013	25.000
15	-TO- 16	-9.994	-17.946	0.111	-1.081	15.000
16	-TO- 1	-40.111	-26.850	0.691	-0.055	120.000
16	-TO- 15	10.105	16.865	0.111	-1.081	15.000
17	-TO- 1	16.591	-13.010	0.430	-0.321	60.000
17	-TO- 5	-82.516*	-4.253	0.980	-4.927	75.000
17	-TO- 8	-7.143	-0.900	0.027	-5.593	10.000
17	-TO- 11	-2.670	0.675	0.010	-1.584	20.000
17	-TO- 12	-2.265	0.290	0.003	-1.377	12.000
17	-TO- 18	18.008*	-2.753	0.265	-0.515	15.000
18	-TO- 17	-17.743*	2.238	0.265	-0.515	15.000
18	-TO- 19	2.745	-7.237	0.045	-1.196	9.000
19	-TO- 1	-0.092	-5.086	0.023	-2.235	50.000
19	-TO- 4	44.190*	-27.418	0.519	0.215	40.000
19	-TO- 5	-56.387	21.490	3.381	7.460	85.000
19	-TO- 18	-2.700	6.041	0.045	-1.196	9.000

20	-TO- 4	-38.701	-7.707	0.604	-0.603	40.000
20	-TO- 21	13.703*	-0.292	0.120	-3.102	12.000
21	-TO- 4	-23.789	-5.616	0.584	-3.976	40.000
21	-TO- 20	-13.584*	-2.810	0.120	-3.102	12.000
21	-TO- 22	17.378	1.428	0.134	-1.897	25.000
22	-TO- 21	-17.244	-3.325	0.134	-1.897	25.000
22	-TO- 23	-10.671	0.366	0.277	-1.177	15.000
22	-TO- 24	7.924	-4.036	0.067	-5.100	7.000
23	-TO- 1	-25.950	-3.447	0.748	-4.231	40.000
23	-TO- 22	10.948	-1.542	0.277	-1.177	15.000
24	-TO- 22	-7.857	-1.063	0.067	-5.100	7.000
24	-TO- 25	-7.131	-3.937	0.029	-2.872	20.000
25	-TO- 1	-32.159	-9.046	0.847	-4.610	90.000
25	-TO- 24	7.160	1.065	0.029	-2.872	20.000

###

###

GENERATORS/SYNCHRONOUS CONDENSERS

###

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	PRES. GEN. MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	245.687	125.868			80.000	-70.000
2	PV	100.000	10.455	150.000	70.000	50.000	-40.000
3	PV	150.000	15.807	200.000	50.000	40.000	-40.000
4	PV	50.000	46.377	60.000	30.000	40.000	-10.000
5	PV	200.000	-8.616	250.000	100.000	70.000	-70.000

STUDY CASE NO.1, "25-BUS SYSTEM" (VOLTAGE ANGLE)

##
COMPLETE LOAD FLOW RESULTS (FINAL STATE)
##

PROGRAM CONVERGED IN 2 ITERATIONS

		BUS VOLT.(KV)		GENERATION		LOAD	
BUS	TYPE	MAGN.	ANGLE	MW	MVAR	MW	MVAR
1	SLK	1.020	0.000	310.377	103.355*	200.000	65.000
2	PV	1.020	3.642	82.766	13.698	10.000	3.000
3	PV	1.010	3.962	134.795	16.788	50.000	17.000
4	PV	1.010	-3.400	60.000	39.762	30.000	10.000
5	PV	1.015	3.259	153.029	-1.141	25.000	8.000
6	PQ	0.990	1.695	0.000	0.000	15.000	5.000
7	PQ	1.000	-0.371	0.000	0.000	15.000	5.000
8	PQ	1.000	-0.870	0.000	0.000	25.000	0.000
9	PQ	0.990	-1.583	0.000	0.000	15.000	5.000
10	PQ	0.999	-0.428	0.000	0.000	15.000	5.000
11	PQ	0.998	-1.169	0.000	0.000	5.000	0.000
12	PQ	0.999	-1.287	0.000	0.000	10.000	0.000

13	PQ	0.988	1.819	0.000	0.000	25.000	8.000
14	PQ	0.958	-3.076	0.000	0.000	20.000	7.000
15	PQ	0.958	-3.807	0.000	0.000	30.000	10.000
16	PQ	0.973	-3.347	0.000	0.000	30.000	10.000
17	PQ	0.999	-1.154	0.000	0.000	60.000	20.000
18	PQ	0.992	-2.599	0.000	0.000	15.000	5.000
19	PQ	1.006	-2.363	0.000	0.000	15.000	5.000
20	PQ	0.988	-5.319	0.000	0.000	25.000	8.000
21	PQ	0.977	-6.291	0.000	0.000	20.000	7.000
22	PQ	0.966	-6.967	0.000	0.000	20.000	7.000
23	PQ	0.988	-3.726	0.000	0.000	15.000	5.000
24	PQ	0.963	-7.866	0.000	0.000	15.000	5.000
25	PQ	0.970	-7.052	0.000	0.000	25.000	8.000

##

LINE		LINE FLOW		LINE LOSSES		FLOW LIM.
		MW	MVAR	MW	MVAR	MW
1	-TO- 3	-22.254	9.043	0.412	-0.200	100.000
1	-TO- 16	47.539	24.568	0.823	0.567	120.000
1	-TO- 17	8.939	3.674	0.097	-1.241	60.000

1	-TO- 19	10.807	-1.285	0.167	-1.860	50.000
1	-TO- 23	29.694	-1.915	0.921	-3.873	40.000
1	-TO- 25	35.652	4.269	0.976	-3.990	90.000
2	-TO- 6	13.349	6.935	0.143	-1.200	100.000
2	-TO- 7	29.868	2.182	0.443	0.533	30.000
2	-TO- 8	29.551	1.580	0.489	0.551	50.000
3	-TO- 1	22.665	-9.243	0.412	-0.200	100.000
3	-TO- 13	27.221	4.746	0.425	0.264	40.000
3	-TO- 14	34.906	4.285	1.445	2.571	40.000
4	-TO- 19	-28.210	19.236	0.228	-0.549	40.000
4	-TO- 20	36.616	7.957	0.533	-0.790	40.000
4	-TO- 21	21.595	2.568	0.471	-4.271	40.000
5	-TO- 10	27.886	-1.095	0.377	-4.053	55.000
5	-TO- 17	62.402	1.097	0.553	-8.665	75.000
5	-TO- 19	37.745	-9.144	1.349	2.116	85.000
6	-TO- 2	-13.206	-8.135	0.143	-1.200	100.000
6	-TO- 13	-1.793	3.135	0.004	-0.381	25.000
7	-TO- 2	-29.425	-1.649	0.443	0.533	30.000
7	-TO- 8	5.430	-1.808	0.017	-0.734	15.000
7	-TO- 12	8.997	-1.540	0.030	-0.957	15.000
8	-TO- 2	-29.062	-1.029	0.489	0.551	50.000
8	-TO- 7	-5.414	1.074	0.017	-0.734	15.000
8	-TO- 9	7.437*	3.055	0.027	-1.041	6.000
8	-TO- 17	2.039	-3.096	0.002	-5.703	10.000
9	-TO- 8	-7.411*	-4.096	0.027	-1.041	6.000
9	-TO- 10	-7.588	-0.904	0.057	-0.682	10.000
10	-TO- 5	-27.509	-2.958	0.377	-4.053	55.000
10	-TO- 9	7.646	0.222	0.057	-0.682	10.000
10	-TO- 11	4.865	-2.261	0.024	-1.285	15.000
11	-TO- 10	-4.841	0.976	0.024	-1.285	15.000
11	-TO- 17	-0.159	-0.976	0.000	-1.606	20.000
12	-TO- 7	-8.967	0.583	0.030	-0.957	15.000
12	-TO- 17	-1.033	-0.582	0.000	-1.385	12.000
13	-TO- 3	-26.797	-4.482	0.425	0.264	40.000
13	-TO- 6	1.797	-3.516	0.004	-0.381	25.000
14	-TO- 3	-33.461	-1.713	1.445	2.571	40.000
14	-TO- 15	13.467	-5.272	0.063	-0.231	25.000
15	-TO- 14	-13.404	5.040	0.063	-0.231	25.000
15	-TO- 16	-16.577*	-15.041	0.134	-1.026	15.000
16	-TO- 1	-46.716	-24.001	0.823	0.567	120.000
16	-TO- 15	16.711*	14.015	0.134	-1.026	15.000
17	-TO- 1	-8.842	-4.914	0.097	-1.241	60.000
17	-TO- 5	-61.849	-9.763	0.553	-8.665	75.000
17	-TO- 8	-2.037	-2.607	0.002	-5.703	10.000
17	-TO- 11	0.159	-0.630	0.000	-1.606	20.000
17	-TO- 12	1.034	-0.803	0.000	-1.385	12.000
17	-TO- 18	11.536	-1.269	0.108	-0.925	15.000
18	-TO- 17	-11.429	0.343	0.108	-0.925	15.000
18	-TO- 19	-3.571	-5.343	0.031	-1.235	9.000
19	-TO- 1	-10.640	-0.576	0.167	-1.860	50.000
19	-TO- 4	28.438	-19.785	0.228	-0.549	40.000
19	-TO- 5	-36.396	11.260	1.349	2.116	85.000

19	-TO- 18	3.602	4.108	0.031	-1.235	9.000
20	-TO- 4	-36.083	-8.748	0.533	-0.790	40.000
20	-TO- 21	11.084	0.749	0.081	-3.202	12.000
21	-TO- 4	-21.124	-6.839	0.471	-4.271	40.000
21	-TO- 20	-11.003	-3.951	0.081	-3.202	12.000
21	-TO- 22	12.130	3.792	0.074	-2.051	25.000
22	-TO- 21	-12.055	-5.843	0.074	-2.051	25.000
22	-TO- 23	-13.324	2.127	0.448	-0.905	15.000
22	-TO- 24	5.388	-3.282	0.031	-5.193	7.000
23	-TO- 1	-28.774	-1.958	0.921	-3.873	40.000
23	-TO- 22	13.772	-3.032	0.448	-0.905	15.000
24	-TO- 22	-5.358	-1.911	0.031	-5.193	7.000
24	-TO- 25	-9.630	-3.090	0.049	-2.810	20.000
25	-TO- 1	-34.676	-8.260	0.976	-3.990	90.000
25	-TO- 24	9.678	0.280	0.049	-2.810	20.000

###

###

GENERATORS/SYNCHRONOUS CONDENSERS

###

LOCATION (BUS)	BUS TYPE	PRES. GEN. MW	PRES. GEN. MVAR	GENERATION CAPACITY MW (MAX.)	MW (MIN.)	MVAR (MAX.)	MVAR (MIN.)
1	SLK	310.377	103.355			80.000	-70.000
2	PV	82.766	13.698	150.000	70.000	50.000	-40.000
3	PV	134.795	16.788	200.000	50.000	40.000	-40.000
4	PV	60.000	39.762	60.000	30.000	40.000	-10.000
5	PV	153.029	-1.141	250.000	100.000	70.000	-70.000

123-6

APPENDIX - C.2

ES OUTPUTS

STUDY CASE NO.1, "14-BUS SYSTEM"

THE EXPERT SYSTEM RESULTS FOR VOLTAGE/VAR ADJUSTMENT
The New Voltages of the Load Buses are as follow:-

BUS: 4	TYPE: PQ	VOLTAGE= 1.0384
BUS: 5	TYPE: PQ	VOLTAGE= 1.0439
BUS: 7	TYPE: PQ	VOLTAGE= 1.0419
BUS: 9	TYPE: PQ	VOLTAGE= 1.0344
BUS: 10	TYPE: PQ	VOLTAGE= 1.0210
BUS: 11	TYPE: PQ	VOLTAGE= 1.0076
BUS: 12	TYPE: PQ	VOLTAGE= 0.9717
BUS: 13	TYPE: PQ	VOLTAGE= 0.9515
BUS: 14	TYPE: PQ	VOLTAGE= 0.9791

The New Q-Gen. of the Gen. Buses are as follow:-

BUS: 3	TYPE: PV	Q-GEN.= 17.3050
BUS: 2	TYPE: PV	Q-GEN.= 0.5363
BUS: 8	TYPE: PV	Q-GEN.= 24.0000
BUS: 6	TYPE: PV	Q-GEN.= 43.4678
BUS: 1	TYPE: SLK	Q-GEN.= 8.4131

The New Settings of Controllers are as follow:-

CONT. SEQ. # 2 LOCATION : 2
CONT. TYPE : GV
PRES. SETTING :1.056882

CONT. SEQ. # 1 LOCATION : 1
CONT. TYPE : GV
PRES. SETTING :1.085646

CONT. SEQ. # 5 LOCATION : 8
CONT. TYPE : GV
PRES. SETTING :1.080987

CONT. SEQ. # 3 LOCATION : 3
CONT. TYPE : GV
PRES. SETTING :1.025198

CONT. SEQ. # 6 LOCATION : 9
CONT. TYPE : SC
PRES. SETTING :0.000000

CONT. SEQ. # 7 LOCATION : 13
CONT. TYPE : SC
PRES. SETTING :9.000000

CONT. SEQ. # 4 LOCATION : 6
CONT. TYPE : GV

PRES. SETTING :1.000000

CONT. SEQ. # 8 LOCATION : 14
CONT. TYPE : SR
PRES. SETTING :0.000000

CONT. SEQ. # 9 LOCATION : 4
CONT. TYPE : TAP
PRES. SETTING :0.975000

CONT. SEQ. # 10 LOCATION : 9
CONT. TYPE : TAP
PRES. SETTING :0.962500

CONT. SEQ. # 11 LOCATION : 5
CONT. TYPE : TAP
PRES. SETTING :0.937500

STUDY CASE NO.2, "30-BUS SYSTEM"

THE EXPERT SYSTEM RESULTS FOR VOLTAGE/VAR ADJUSTMENT
The New Voltages of the Load Buses are as follow:-

BUS: 3	TYPE: PQ	VOLTAGE= 1.0456
BUS: 4	TYPE: PQ	VOLTAGE= 1.0356
BUS: 6	TYPE: PQ	VOLTAGE= 1.0267
BUS: 7	TYPE: PQ	VOLTAGE= 1.0207
BUS: 9	TYPE: PQ	VOLTAGE= 1.0479
BUS: 10	TYPE: PQ	VOLTAGE= 1.0295
BUS: 12	TYPE: PQ	VOLTAGE= 1.0320
BUS: 14	TYPE: PQ	VOLTAGE= 1.0138
BUS: 15	TYPE: PQ	VOLTAGE= 0.9996
BUS: 16	TYPE: PQ	VOLTAGE= 1.0238
BUS: 17	TYPE: PQ	VOLTAGE= 1.0224
BUS: 18	TYPE: PQ	VOLTAGE= 0.9690
BUS: 19	TYPE: PQ	VOLTAGE= 0.9541
BUS: 20	TYPE: PQ	VOLTAGE= 0.9662
BUS: 21	TYPE: PQ	VOLTAGE= 1.0143
BUS: 22	TYPE: PQ	VOLTAGE= 1.0140
BUS: 23	TYPE: PQ	VOLTAGE= 0.9750
BUS: 24	TYPE: PQ	VOLTAGE= 0.9897
BUS: 25	TYPE: PQ	VOLTAGE= 0.9809
BUS: 26	TYPE: PQ	VOLTAGE= 0.9626
BUS: 27	TYPE: PQ	VOLTAGE= 0.9845
BUS: 28	TYPE: PQ	VOLTAGE= 1.0245
BUS: 29	TYPE: PQ	VOLTAGE= 0.9638
BUS: 30	TYPE: PQ	VOLTAGE= 0.9518

The New Q-Gen. of the Gen. Buses are as follow:-

BUS: 11	TYPE: PV	Q-GEN.= 27.7957
BUS: 2	TYPE: PV	Q-GEN.= 32.3548
BUS: 8	TYPE: PV	Q-GEN.= 43.1120
BUS: 13	TYPE: PV	Q-GEN.= 50.0000
BUS: 5	TYPE: PV	Q-GEN.= 39.3192
BUS: 1	TYPE: SLK	Q-GEN.= 1.5908

The New Settings of Controllers are as follow:-

CONT. SEQ. # 6 LOCATION : 13
CONT. TYPE : GV
PRES. SETTING :1.095274

CONT. SEQ. # 3 LOCATION : 5
CONT. TYPE : GV
PRES. SETTING :1.030777

CONT. SEQ. # 5 LOCATION : 11
CONT. TYPE : GV

123-6

```
  PRES. SETTING :1.100000

CONT. SEQ. # 4  LOCATION -:  8
CONT. TYPE :  GV
  PRES. SETTING :1.028691

CONT. SEQ. # 2  LOCATION :  2
CONT. TYPE :  GV
  PRES. SETTING :1.062143

CONT. SEQ. # 1  LOCATION :  1
CONT. TYPE :  GV
  PRES. SETTING :1.094200

CONT. SEQ. # 15  LOCATION :  28
CONT. TYPE :  TAP
  PRES. SETTING :0.962500

CONT. SEQ. # 14  LOCATION :  10
CONT. TYPE :  TAP
  PRES. SETTING :0.962500

CONT. SEQ. # 12  LOCATION :  4
CONT. TYPE :  TAP
  PRES. SETTING :0.937500

CONT. SEQ. # 13  LOCATION :  6
CONT. TYPE :  TAP
  PRES. SETTING :1.012500

CONT. SEQ. # 9  LOCATION :  24
CONT. TYPE :  SC
  PRES. SETTING :8.600000

CONT. SEQ. # 8  LOCATION :  15
CONT. TYPE :  SC
  PRES. SETTING :25.000000

CONT. SEQ. # 7  LOCATION :  10
CONT. TYPE :  SC
  PRES. SETTING :19.000000

CONT. SEQ. # 10  LOCATION :  28
CONT. TYPE :  SR
  PRES. SETTING :0.000000

CONT. SEQ. # 11  LOCATION :  16
CONT. TYPE :  SR
  PRES. SETTING :0.000000
```

STUDY CASE NO.1, "6-BUS SYSTEM"

EXPERT SYSTEM RESULTS FOR GENERATION RESCHEDULING

The New Values of the Power Flows are as follow:-

LINE NO. :1	BETWEEN : 1	- 2	P. FLOW : 4.9217	P. FLOW LIM. : 33.5000
LINE NO. :2	BETWEEN : 1	- 4	P. FLOW : 27.0423	P. FLOW LIM. : 36.0000
LINE NO. :3	BETWEEN : 1	- 5	P. FLOW : 25.0961	P. FLOW LIM. : 30.0000
LINE NO. :4	BETWEEN : 2	- 3	P. FLOW : 10.0545	P. FLOW LIM. : 15.5000
LINE NO. :5	BETWEEN : 2	- 4	P. FLOW : 49.4528	P. FLOW LIM. : 60.0000
LINE NO. :6	BETWEEN : 2	- 5	P. FLOW : 21.9075	P. FLOW LIM. : 30.0000
LINE NO. :7	BETWEEN : 2	- 6	P. FLOW : 32.3859	P. FLOW LIM. : 40.0000
LINE NO. :8	BETWEEN : 3	- 5	P. FLOW : 20.4684	P. FLOW LIM. : 23.0000
LINE NO. :10	BETWEEN : 4	- 5	P. FLOW : 4.2485	P. FLOW LIM. : 30.0000
LINE NO. :9	BETWEEN : 3	- 6	P. FLOW : 40.0000	P. FLOW LIM. : 40.0000
LINE NO. :11	BETWEEN : 5	- 6	P. FLOW : -0.6303	P. FLOW LIM. : 15.0000

The New Generation Schedule is as follow:-

GEN. LOCAT. :3 PR. SET : 50.4864

GEN. LOCAT. :2 PR. SET : 108.3078

STUDY CASE NO.2, "25-BUS SYSTEM"

EXPERT SYSTEM RESULTS FOR GENERATION RESCHEDULING

The New Values of the Power Flows are as follow:-

LINE NO. :16	BETWEEN : 5	- 17	P. FLOW : 69.7797	P. FLOW LIM. : 75.0000
LINE NO. :15	BETWEEN : 5	- 10	P. FLOW : 30.0423	P. FLOW LIM. : 55.0000
LINE NO. :14	BETWEEN : 4	- 21	P. FLOW : 21.1814	P. FLOW LIM. : 40.0000
LINE NO. :13	BETWEEN : 4	- 20	P. FLOW : 36.2128	P. FLOW LIM. : 40.0000
LINE NO. :12	BETWEEN : 4	- 19	P. FLOW : -37.3873	P. FLOW LIM. : 40.0000
LINE NO. :11	BETWEEN : 3	- 14	P. FLOW : 38.2654	P. FLOW LIM. : 40.0000
LINE NO. :10	BETWEEN : 3	- 13	P. FLOW : 31.0880	P. FLOW LIM. : 40.0000
LINE NO. :9	BETWEEN : 2	- 8	P. FLOW : 27.4219	P. FLOW LIM. : 50.0000
LINE NO. :7	BETWEEN : 2	- 6	P. FLOW : 9.2942	P. FLOW LIM. : 100.0000
LINE NO. :6	BETWEEN : 1	- 25	P. FLOW : 36.0076	P. FLOW LIM. : 90.0000
LINE NO. :5	BETWEEN : 1	- 23	P. FLOW : 30.0786	P. FLOW LIM. : 40.0000
LINE NO. :4	BETWEEN : 1	- 19	P. FLOW : 10.6586	P. FLOW LIM. : 50.0000
LINE NO. :3	BETWEEN : 1	- 17	P. FLOW : 5.0406	P. FLOW LIM. : 60.0000
LINE NO. :2	BETWEEN : 1	- 16	P. FLOW : 44.3736	P. FLOW LIM. : 120.0000
LINE NO. :1	BETWEEN : 1	- 3	P. FLOW : -29.9775	P. FLOW LIM. : 100.0000
LINE NO. :35	BETWEEN : 24	- 25	P. FLOW : -9.9723	P. FLOW LIM. : 20.0000
LINE NO. :34	BETWEEN : 22	- 24	P. FLOW : 5.0319	P. FLOW LIM. : 7.0000
LINE NO. :33	BETWEEN : 22	- 23	P. FLOW : -13.7011	P. FLOW LIM. : 15.0000
LINE NO. :32	BETWEEN : 21	- 22	P. FLOW : 11.3708	P. FLOW LIM. : 25.0000
LINE NO. :31	BETWEEN : 20	- 21	P. FLOW : 10.6985	P. FLOW LIM. : 12.0000
LINE NO. :30	BETWEEN : 18	- 19	P. FLOW : -1.4002	P. FLOW LIM. : 9.0000
LINE NO. :29	BETWEEN : 17	- 18	P. FLOW : 13.7353	P. FLOW LIM. : 15.0000
LINE NO. :28	BETWEEN : 15	- 16	P. FLOW : -13.4987	P. FLOW LIM. : 15.0000
LINE NO. :27	BETWEEN : 14	- 15	P. FLOW : 16.5683	P. FLOW LIM. : 25.0000
LINE NO. :26	BETWEEN : 12	- 17	P. FLOW : -2.1272	P. FLOW LIM. : 12.0000
LINE NO. :8	BETWEEN : 2	- 7	P. FLOW : 27.9505	P. FLOW LIM. : 30.0000
LINE NO. :25	BETWEEN : 11	- 17	P. FLOW : 0.4874	P. FLOW LIM. : 20.0000
LINE NO. :24	BETWEEN : 10	- 11	P. FLOW : 5.5136	P. FLOW LIM. : 15.0000
LINE NO. :23	BETWEEN : 9	- 10	P. FLOW : -9.0198	P. FLOW LIM. : 10.0000
LINE NO. :22	BETWEEN : 8	- 17	P. FLOW : 0.7029	P. FLOW LIM. : 10.0000
LINE NO. :21	BETWEEN : 8	- 9	P. FLOW : 6.0000	P. FLOW LIM. : 6.0000
LINE NO. :20	BETWEEN : 7	- 12	P. FLOW : 7.8903	P. FLOW LIM. : 15.0000
LINE NO. :19	BETWEEN : 7	- 8	P. FLOW : 4.6900	P. FLOW LIM. : 15.0000
LINE NO. :18	BETWEEN : 6	- 13	P. FLOW : -5.6876	P. FLOW LIM. : 25.0000
LINE NO. :17	BETWEEN : 5	- 19	P. FLOW : 45.2163	P. FLOW LIM. : 85.0000

The New Generation Schedule is as follow:-

GEN. LOCAT. :5 PR. SET : 170.0282

GEN. LOCAT. :2 PR. SET : 74.6583

GEN. LOCAT. :4 PR. SET : 50.0000

GEN. LOCAT. :3 PR. SET : 150.0000

STUDY CASE NO.1, "25-BUS SYSTEM" (VOLTAGE ANGLE)

EXPERT SYSTEM RESULTS FOR VOLTAGE ANGLE ADJUSTMENT

The New Voltage Angles of the Generator Buses are as follow:-

BUS NAME :5	BUS TYPE : PV	VOLTAGE ANGLE : 3.3513
BUS NAME :4	BUS TYPE : PV	VOLTAGE ANGLE : -3.3083
BUS NAME :3	BUS TYPE : PV	VOLTAGE ANGLE : 4.0000
BUS NAME :2	BUS TYPE : PV	VOLTAGE ANGLE : 3.7222
BUS NAME :1	BUS TYPE : SLK	VOLTAGE ANGLE : 0.0000

The New Settings of Generators are as follow:-

GEN. LOCAT. :3	PR. SETT. : 134.7947	MAX. GEN. : 200.0000	MIN. GEN. : 50.0000
GEN. LOCAT. :4	PR. SETT. : 60.0000	MAX. GEN. : 60.0000	MIN. GEN. : 30.0000
GEN. LOCAT. :2	PR. SETT. : 82.7664	MAX. GEN. : 150.0000	MIN. GEN. : 70.0000
GEN. LOCAT. :5	PR. SETT. : 153.0287	MAX. GEN. : 250.0000	MIN. GEN. : 100.0000

NOMENCLATURE

AI	=	Artificial Intelligence
CLIPS	=	'C' Language Integrated Production Systm
EMS	=	Energy Management System
ES	=	Expert System
IAP	=	Intelligence Alarm Process
IE	=	Inference Engine
I/O	=	Input/Output
KB	=	Knowledge Base
OPS5	=	Official Production System Version 5
MEC	=	Most Efficient Controller
RPC	=	Reactive Power Control
SCADA	=	Supervisory Control and Data Acquisition
SMEC	=	Second Most Efficient Controller

REFERENCES

1) Chen-Ching Liu and Paul Vanson, "Expert Systems: Development Experience and User Requirements", Electra, No. 146, pp. 29-67, Feb. 1993.

2) CIGRE: Task Force 39.03 of Study Committee 38, "Expert System Applied to Voltage and VAR Control Final Report", Electra, No. 139, PP. 109-131, Dec. 1991.

3) Zhang, Z.Z. et al., "Expert Systems in Electric Power System - A Bibliographical Survey:, IEEE Trans. on Power Systems, Vol. 4, No. 4, pp. 1355-1362, Oct. 1988.

4) CIGRE: Task Force 39.03, "Exploring User Requirements of Expert Systems in Power System Operation and Control", Electra, No. 146, pp. 69-87, Feb. 1993.

5) CIGRE: Working Group 02 of Study Committee 38, "An International Survey of the Present Status and the Perspective of Expert Systems on Power Analysis and Techniques", Electra, No. 123, pp. 72-93, 1988.

6) Cheng, S.J. Malik, O.P. Hope, G.S., "An Expert System for Voltage and Reactive Control of a Power System", IEEE Trans. on Power Systems, Vol. 3, No. 4, pp. 1449-1455, Nov. 1988.

7) Dillon, T.S. and Laughton, M.A., Expert System Applications in Power Systems, Prentice Hall Int. Ltd., U.K., 1990.

8) Liu, C.C. and Tomsovic, K.L., "An Expert System Assisting Decision-Making of Reactive Power/Voltage Control", IEEE Trans. on Power Systems, Vol. PWRS-1, No. 3, pp. 195-201, Aug. 1986.

9) Machias, A.V., Dialynas, E.N., Protopapas, C.A., "An Expert System Approach to Designing and Testing Substation Grounding Grids", IEEE Trans. on Power

Delivery, Vol. 4, No. 1, pp. 234-239, Jan. 1989.

10) Fukui, C. and Kawakami, J., "An Expert System for Fault Section Estimation Using Information from Protective Relays and Circuit Breakers", IEEE Trans. on Power Delivery, Vol. 4, pp. 83-90, Oct. 1986.

11) Cardozo, Eleri and Talukdar, S.M. "A Distributed Expert System for Fault Diagnosis", IEEE Trans. on Power Systems, Vol. 3, No. 2, pp. 641-646, May 1988.

12) Talukdar, S.N. and Others, "The Operator's Assistant - An Intelligent, Expandable Program for Power System Trouble Analysis", IEEE Trans. on Power Systems, Vol.PWRS-1, No. 3, pp. 182-187, Aug. 1986.

13) Amelink, H. and Others, "Dispatcher Alarm and Message Proceedings", IEEE Trans. on Power Systems, Vol. PWRS-1, No. 3, pp. 182-187, Aug. 1986.

14) Schulte,R.P. and Others, "Artificial Intelligence Solutions to Power System Operating Problems", IEEE Trans. on Power Systems, Vol. PWRS-2, No. 4, pp. 920-926, Nov. 1987.

15) Salah, Mazen A.A., The Development of An Intelligent Alarm Processor, Master Thesis, Electrical Engineering Department-KFUPM, July 1991.

16) Tomsovic, Kevin and Others, "An Expert System As A Dispatchers' Aid for the Isolation of the Section Faults", IEEE Trans. on Power Delivery, Vol. PWRS-2, No. 3, pp. 736-743, July 1987.

17) Liu, Chen-Ching and Others, "An Expert System Operational Aid for Restoration and Loss Reduction of Distribution Systems", IEEE Trans. on Power Systems, Vol. 3, No. 2, pp. 619-626, May 1988.

18) Russell, B. Don and Watson, Karan, "Power Substation Automation Using A Knowledge Based System - Justification and Preliminary Field Experiments", IEEE Trans. on Power Delivery, Vol. PWRD-2, No. 4, pp. 1090-1096, Oct. 1987.

19) Fijiwara, R. and Others, "An Intelligent Load Flow Engine for Power System Planning", IEEE Trans. on Power Systems, Vol. PWRS-1, No. 3, pp. 302-307, Aug. 1986.

20) A. Al-Shehri, M. Guizani, C. Belhaj, and M. Aref, "AI Technique for Load Flow Planning and Contingency Analysis", Proceedings of IEEE International Conference (MELCON 91), Yugoslavia, May 1991.

21) A. Al-Shehri, C. Belhaj, M. Aref, and M. Guizani, "An Expert System for Load Flow Analysis Planning", Proceedings of 4th Symposium on Expert Systems Application to Power Systems, Australia, January, 1993.

22) Sobajic, D. and Y. Pao, "An Artificial Intelligence System for Power System Contingency Screening", IEEE Trans. on Power Apparatus and Systems, Vol. 3, No. 2, pp. 647-653, May 1988.

23) Godard, T.F. and Puttgen, H.B., "A Reactive Path Concept Applied Within a Voltage Control Expert System", IEEE Trans. on Power Systems, Vol. 6, No. 2, pp. 787-793, May 1991.

24) Joseph Giarratano and Gary Riley, Expert Systems Principles and Programming, PWS-KENT Publishing Co;, Boston, 1989.

25) CLIPS Basic Programming Guide, Version, 6.0, NASA Lyndon B. Johnson Space Center, June, 1993.

26) G.T. Heydt, Computer Analysis Methods for Power Systems, McMillan Publishing Co., New York, 1986.

27) Mamandur, K.R.C. and Berg, G.J., "Economic Shift in Electric Power Generation with Line Flow Constraints", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-97, No. 5, pp. 1618-1626, Sept./Oct. 1978.

28) Hano, I., et al., "Real Time Control of System Voltage and Reactive Power", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-88, No. 10, pp. 1544-

1559, Oct. 1969.

29) Lai, K.C, Lee, W.J and Chen, M.S., "Design of a Microcomputer Based Operator Assistance System for Real Time Voltage and Reactive Power Connection", IEEE Trans. on Power Systems, Vol.6, No. 2, pp. 723-728, May 1991.

30) Iba, Kenji and Others, "Practical Reactive Power Allocation/Operating Planning Using Successive Linear Programming", IEEE Trans. on Power Systems, Vol. 3, No. 2, pp. 558-566, May 1988.

31) Mamandur, K.R.C. and Chenoweth, R.D., "Optimal Control of Reactive Power Flow for Improvements in Voltage Profiles and for Real Power Loss Minimization", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-100, No. 7, pp. 3185-3194, July 1981.

32) Mamandur, K.R.C., "Emergency Adjustments to VAR Control Variables to Alleviate Over-Voltages, Under-Voltages, and Generator VAR Limit Violations", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-101, No. 5, pp. 1040-1047, May 1982.

33) L.L. Ferris, and A.M. Sasson, "Investigation of the Load-Flow Problem", Proc. IEE, Vol. 115, No. 10, pp. 1459-1470, Oct. 1968.

34) Allen J. Wood and Bruse F. Wollenberg, Power Generation, Operation, and Control, John Wiley & Sons, Inc., Canada, 1984.

35) R.B. Gungor, N.F. Tsang and B. Webb, "A Technique for Optimizing Real and Reactive Power Schedules", IEEE Trans. on Power Apparatus and Systems, Vol. PAS-90, pp. 1781-1790, 1971